applied sciences

MDPI

Article

Transfer Learning from Deep Neural Networks for Predicting Student Performance

Maria Tsiakmaki, Georgios Kostopoulos, Sotiris Kotsiantis * and Omiros Ragos

Department of Mathematics, University of Patras, 26504 Rio Patras, Greece; m.tsiakmaki@gmail.com (M.T.); kostg@sch.gr (G.K.); ragos@math.upatras.gr (O.R.)

* Correspondence: sotos@math.upatras.gr

Received: 14 February 2020; Accepted: 19 March 2020; Published: 21 March 2020

Abstract: Transferring knowledge from one domain to another has gained a lot of attention among scientists in recent years. Transfer learning is a machine learning approach aiming to exploit the knowledge retrieved from one problem for improving the predictive performance of a learning model for a different but related problem. This is particularly the case when there is a lack of data regarding a problem, but there is plenty of data about another related one. To this end, the present study intends to investigate the effectiveness of transfer learning from deep neural networks for the task of students' performance prediction in higher education. Since building predictive models in the Educational Data Mining field through transfer learning methods has been poorly studied so far, we consider this study as an important step in this direction. Therefore, a plethora of experiments were conducted based on data originating from five compulsory courses of two undergraduate programs. The experimental results demonstrate that the prognosis of students at risk of failure can be achieved with satisfactory accuracy in most cases, provided that datasets of students who have attended other related courses are available.

Keywords: transfer learning; deep learning; educational data mining; student performance prediction

1. Introduction

Transferring knowledge from one domain to another has gained a lot of attention among scientists in the past few years. Consider the task of predicting student performance (pass/fail) in higher education courses. According to the traditional supervised learning approach, a sufficient amount of training data, regarding a specific course C_A, is required for building an accurate predictive model which is subsequently used for making predictions on testing data derived from the same course. If the testing dataset is derived from a different course, C_B, sharing some common characteristics with course C_A (hereinafter referred to as related or similar courses), then transfer learning is the appropriate machine learning methodology for building accurate learning models in a more efficient manner, since it could contribute to the improvement of the predictive performance of the target domain model (course C_B) exploiting the knowledge of the source domain (course C_A) [1]. In a nutshell, a learning model is trained for a specific task using data derived from a source domain and, subsequently, it is reused for another similar task in the same domain or the same task in a different domain (target domain) [2,3]. More generally, when we lack information about a problem, we could train a learning model for a related problem, for which there is plenty of information, and apply it to the existing one.

Transfer learning is currently gaining popularity in deep learning [4]. Not long ago, it was claimed as the second "driver of machine learning commercial success", whereas supervised learning was the first one [5]. Pre-trained deep networks, usually trained on large datasets and thus requiring significant computation time and resources, are employed as the starting point for other machine learning problems due to their ability to be repurposed either for a new or for a similar task. Therefore,

these networks could support complex problems in a more efficient way, since they can decrease the training time for building a new learning model and finally improve its generalization performance [6].

In recent years, several types of Learning Management Systems (LMSs) have been successfully adopted by universities and higher education institutions, recording a variety of student learning features and gathering huge amounts of educational data. Educational Data Mining (EDM) is a fast-growing scientific field offering the potential to analyze these data and harness valuable knowledge from them. To this end, a plethora of predictive algorithms have been effectively applied in educational contexts for solving a wide range of problems [7]. However, building predictive models in the EDM field through transfer learning methods has been poorly studied so far. Therefore, the main question in the present study is whether a predictive model trained on a past course would perform well on a new one. Boyer and Veeramachaneni observe that courses (a) might evolve over time in a dissimilar way, even if they are not much different in terms of context and structure, (b) are populated with different students and instructors, and (c) might have features that cannot be transferred (e.g., a feature defined on a specific learning resource which is not available on another course) [8]. In addition, the complexity of LMSs as well as the course design have a significant impact on the course progress during the semester [9]. Therefore, there may be problems where transfer learning might not reflect the anticipating results, showing some uncertainty about the predictive accuracy of the newly created learning model [10].

In this context, the present study aims to propose a transfer learning methodology for predicting student performance in higher education, a task that has been extensively studied in the field of EDM through traditional supervised methods. To this purpose, we exploit a set of five datasets corresponding to five undergraduate courses, each one lasting one semester, all supported by a Moodle platform. Initially, we form all the unique pairs of datasets (twenty pairs in total) matching the features of the paired courses one by one and generating new features if necessary. Next, a deep network model is trained by using the dataset of the first course and, subsequently, it is applied on the dataset of the second course for further training after a predefined number of epochs. Deep networks have been successfully applied in the EDM field for solving important educational problems, such as predicting student performance [11–14], dropout [15–17], or automatic feature extraction [18]. The main objective is to discover whether transfer learning accelerates training and improves the predictive performance utilizing the potential of deep neural networks in the EDM field. On this basis, we hope to provide a useful contribution for researchers.

The remainder of this paper is organized as follows. In the next section, we discuss the transfer learning approach, while in Section 3 we present an overview of some related studies in the EDM field. The research goal, together with an analysis of the datasets and description of the proposed transfer learning method, is set in Section 4. The experimental results are presented in Section 5, while Section 6 discusses the research findings. Finally, Section 7 summarizes the study, considering some thoughts for future work.

2. The Transfer Learning Approach

The traditional supervised learning methods exploit labeled data to obtain predictive models in the most efficient way. Let us consider the task of predicting whether a student is going to successfully pass or fail the examinations of an undergraduate course C_A. In this case, the training and the testing set are both derived from the same domain (course). The training set is used to build a learning model h by means of a classification algorithm (e.g., a deep network) and subsequently it is applied on the testing set for evaluating its predictive performance (Figure 1a). Some key requirements for achieving high performance models are the quality and sufficiency of the training data which are, unfortunately, not always easy to meet in real world problems. In addition, the direct implementation of model h for a different course C_B or a new task (e.g., predicting whether a student is going to drop out of the course) seems rather difficult. The existing model does not have the ability to generalize well to data

coming from a different distribution, while, at the same time, it is not applicable, since the class labels of the two tasks are different.

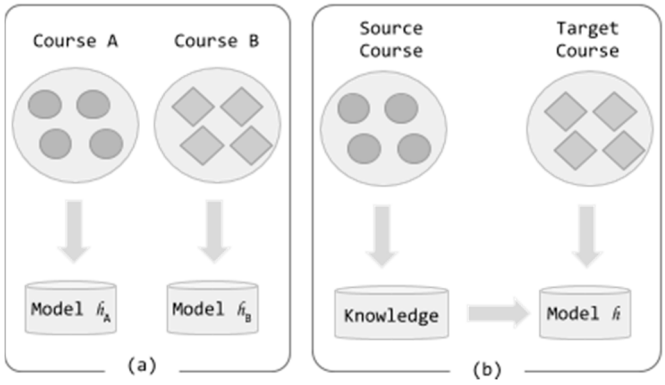

Figure 1. The traditional machine learning process (**a**), the transfer learning process (**b**).

Contrasting these methods, knowledge transfer or transfer learning intends to improve the performance of learning and provide efficient models in cases where data sources are limited or difficult and expensive to acquire [1,2], primarily due to their generalization ability to heterogeneous data (i.e., data from different domains, tasks and distributions [19]). Transfer learning might help us to train a predictive model h based on data derived from course C_A (source course) and apply it on data derived from a different but related course C_B (target course), which are not sufficient to train a model, for predicting the performance of a student. This indeed, is the aim of transfer learning: transfer the knowledge acquired from course C_A to course C_B and improve the predictive performance of model h (Figure 1b) instead of developing a totally new model, on the basis that both datasets should share some common attributes (i.e., common characteristics of students, such as their academic achievements or interactions within an LMS).

More formally, the transfer learning problem is defined as follows [1,20]:

A domain $\mathcal{D}$ is formed by a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$. A learning task $\mathcal{T}$ is formed by a label space Y and an objective predicted function $f(\cdot)$. The function f can be also written as $P(y|x)$, representing the conditional probability distribution of label y given a new instance x. $P(y|x)$ is learned from the training data $\{\mathcal{X}, \mathrm{Y}\}$. Given a source domain $\mathcal{D}_S = \{\mathcal{X}_S, P_S(X)\}$, its corresponding learning task $\mathcal{T}_S = \{\mathrm{Y_S},\ f_S(\cdot)\}$, a target domain $\mathcal{D}_T = \{\mathcal{X}_T, P_S(X)\}$ and its corresponding learning task $\mathcal{T}_T = \{\mathrm{Y_T},\ f_T(\cdot)\}$, the purpose of transfer learning is to obtain an improved target predictive function $f_T(\cdot)$ by using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S \neq \mathcal{T}_T$. The fact that $\mathcal{D}_S \neq \mathcal{D}_T$ means that either $\mathcal{X}_S \neq \mathcal{X}_T$ or $P(X_S) \neq P(X_T)$, where $X_{S_i} \in \mathcal{X}_S$ and $X_{T_i} \in \mathcal{X}_T$. Similarly, the fact that $\mathcal{T}_S \neq \mathcal{T}_T$ means that either $\mathrm{Y_S} \neq \mathrm{Y_T}$ or $f_S(\cdot) \neq f_T(\cdot)$.

The inequalities contained in the definition form four different transfer learning settings:

- $\mathcal{X}_S \neq \mathcal{X}_T$: the feature space of the source and target domain are different. For example, the courses have different structure and context;
- $P(X_S) \neq P(X_T)$: the marginal probability distribution of the source and target domain are different. For example, the same courses offered by different departments or the same course offered in different years by the same department, thus consisting of different students;
- $\mathrm{Y_S} \neq \mathrm{Y_T}$: the label spaces of the source and target task are different (this setting usually occurs with setting four). For example, the source domain has two classes (e.g., {pass, fail}) and the target domain has six classes (e.g., {A, B, C, D, E, F});
- $f_S(\cdot) \neq f_T(\cdot)$: the conditional probability distributions of the source and target tasks are different. For example, the source and target courses are very unbalanced in relation to the defined classes.

Based on the above definition and conditions, three types of transfer learning settings are identified [1,8,18,21]: inductive transfer learning, transductive transfer learning and unsupervised

transfer learning. In inductive transfer learning, the source domain is different but related to the target domain ($\mathcal{D}_S \neq \mathcal{D}_T$) regardless of the relationship between the tasks. In transductive transfer learning, both source and target task are the same ($\mathcal{T}_S = \mathcal{T}_T$), while the domains are different ($\mathcal{D}_S \neq \mathcal{D}_T$). Finally, in unsupervised transfer learning, the tasks are different ($\mathcal{T}_S \neq \mathcal{T}_T$), while both datasets do not contain labels. The latter type is intended for clustering and dimensionality reduction tasks.

3. Related Work

Predicting students' learning outcomes is considered one of the major tasks of the EDM field [22]. This is demonstrated by a great number of significant studies which put emphasis on the development and implementation of data mining methods and machine algorithms for resolving a plethora of predictive problems [23]. These problems are mainly intended to predict the future value of an attribute (e.g., students' grades, academic performance, dropout, etc.) based on a set of input attributes that describe a student. One typical problem is to detect whether a student is going to successfully pass or fail a course by the end of a semester based on his/her activity on the LMS, as in this study. The successful and accurate detection of students at risk of failure is of vital importance for educational institutions, since remedial measures and intervention strategies could be applied to support low performers and enhance their overall learning performance [24]. It is therefore necessary to build very accurate and robust learning models. Transfer learning could contribute to improving these models, since prior knowledge regarding a specific task could be useful to another similar task. Transfer learning is an approach which has still not been sufficiently examined in the field of EDM, as evidenced by the study of the current literature. To the best of our knowledge, there are few studies focusing on resolving prediction problems through transferring learning models from one domain to another, although this prospect is appealing. These studies indicate that building models based on a particular course and them applying to a new one (different but somehow related) is a rather complex task, which, unfortunately, does not always reflect the anticipating outcomes [10]. A list of some notable works regarding transfer learning in the EDM field are presented in the following paragraphs.

Ding et al. investigated the transferability of dropout prediction across Massive Online Open Courses (MOOCs) [9]. Therefore, they presented two variations of transfer learning based on autoencoders: (a) using the transductive principal component analysis, and (b) adding a correlation alignment loss term. The input data were click-stream log events of mixtures of similar and dissimilar courses. The proposed transfer learning methods proved to be quite effective for improving the dropout prediction, in terms of Area Under Curve (AUC) scores, compared to the baseline method. In a similar study, Vitiello et al. [25] examined how models trained on a MOOC system could be transferred to another. Therefore, they built a unified model allowing the early prediction of dropout students across two different systems. At first, the authors confirmed significant differences between the two systems, such as the number of active students and the structure of courses. After that, they defined a set of features based on the event logs of the two systems. Overall, three dropout prediction experiments were conducted: one for each separate system, one where each system applied a learning model built on the other system and one where the dataset contained data from both systems. The accuracy measure was above the baseline threshold (0.5) in most cases.

The method put forward by Hunt et al. [26] examined the effectiveness of TrAdaBoost, an extended AdaBoost version in the transfer learning framework, for predicting students' graduation rates in undergraduate programs. The dataset was based on a set of academic and demographic features (152 features in total) regarding 7637 students of different departments. Two separate experiments were conducted, each time using specific data for the training set. In the first experiment, the training set comprised all students apart from those studying engineering, while in the second one, the training set comprised all students that were suspended on academic warnings. The experimental results showed that the TrAdaBoost method recorded the smallest error in both cases. In the same context, Boyer and Veeramachaneni suggested two different approaches for predicting student dropout taking into account the selection method of the training data and how to make use of past courses information [8].

Therefore, several tests were performed using either all available information for a learner or a fixed subset of them. In addition, two different scenarios were formulated: inductive and transductive transfer learning. The experimental results indicated that the produced learning models did not always perform as intended. Very recently, Tri, Chau and Phung [27] proposed a transfer learning algorithm, named CombinedTL, for the identification of failure-prone students. Therefore, they combined a case-based reasoning framework and four instance-based transfer learning algorithms (MultiSource, TrAdaboost, TrAdaboost, and TransferBoost). The experimental results showed that the proposed method outperformed the single instance-based transfer learning algorithms. In addition, the authors compared the CombinedTL with typical case retrieval methods (k-NN and C4.5), experimenting with a varying number of target instances, finding that the performance of the proposed method was improved as the number of target instances was increased.

The notion of domain adaptation is highly associated with transfer learning. Zeng et al. [28] proposed a self-training algorithm (DiAd) which adjusts a classifier trained on the source domain to the target domain based on the most confident examples of the target domain and the most dissimilar examples of the source domain. Moreover, the classifier is adjusted to the new domain without using any labeled examples. Very recently, López-Zambrano et al. [29] investigated the portability of learning models based on Moodle log data regarding the courses of different universities. The authors explored whether the grouping of similar courses (i.e., similarity level of learning activities) influence the portability of the prediction models. The experimental results showed that models based on discretized datasets obtained better portability than those based on numerical ones.

4. Research Methodology

4.1. Research Goal

The main purpose of our study is to evaluate the effectiveness of transfer learning methods in the EDM field. More specifically, we investigate whether a deep learning model that has been trained using student data from one course can be repurposed for other related courses. Deep neural networks are represented by a number of connecting weights between the layers. During the training process, these weights are adjusted in order to minimize the error of the expected output. Therefore, the main notion behind the suggested transfer learning approach is to initialize a deep network using the pre-tuned weights from a similar course. Two main research questions guide our research:

(1) Can the weights of a deep learning model trained on a specific course be used as the starting point for a model of another related course?

(2) Will the pre-trained model reduce the training effort for the deep model of the second course?

4.2. Data Analysis

In the present study, we selected data regarding five compulsory courses of two undergraduate programs offered by the Aristotle University of Thessaloniki in Greece. More precisely, three courses (Physical Chemistry I (Spring 2018) and Analytical Chemistry Laboratory (Spring 2018, Spring 2019)) were offered by the department of Chemical Engineering, while two courses (Physics III (Spring 2018, Spring 2019)) were offered by the department of Physics. Table 1 provides detailed information regarding the gender and target class distribution of the five courses.

Table 1. Gender and target class distribution of the courses.

Course	Female		Male		Pass		Fail	
C1: Physical Chemistry I (Spring 2018)	122	43.3%	160	56.8%	134	47.5%	148	52.5%
C2: Physics III (Spring 2018)	90	50.0%	90	50.0%	74	41.1%	106	56.9%
C3: Analytical Chemistry Lab (Spring 2018)	57	48.2%	72	55.8%	105	81.4%	24	18.6%
C4: Physics III (Spring 2019)	80	50.6%	78	49.4%	68	43.0%	90	57.0%
C5: Analytical Chemistry Lab (Spring 2019)	61	52.1%	56	47.9%	100	85.5%	17	14.5%

Each course was supported by an online LMS, embedding a plethora of resources and activities. The course pages were organized into topic sections containing the learning material in the form of web pages, document files and/or URLs, while the default announcements forum was enabled for each course allowing students to post threads and communicate with colleagues and tutors. Each course required the submission of several assignments, which were evaluated on a grading scale from zero to 10. All sections were available to the students until the end of the semester, while the course final grade corresponded to the weighted average of the marks of all submitted assignments and the finishing exam. Note that successful completion of the course required a minimum grade of five.

For the purpose of our study, the collected datasets comprised six different types of learning resources: forums, pages, recourses, folders, URLs and assignments (Table 2). For example, course C_1 was associated with one forum, seven pages, 17 resources, two folders and eight assignments, three of which were compulsory. Regarding the forum module, we recorded the total number of views for each student. We also recorded the total number of times students accessed a page, a resource, a folder or a URL. Moreover, two counters were embedded in the course LMS, aggregating the number of student views (course total views) as well as the number of every type of recorded activity for a student (course total activity). Learning activities that were not accessed by students were not included in the experiments, while a student who did not access a learning activity was marked with a zero score. Finally, a custom Moodle plugin was developed, enabling the creation of the five datasets [30].

Table 2. Features extracted from students' low-level interaction logs.

Learning Resources	C_1	C_2	C_3	C_4	C_5	Description	Possible Values
Forum	1	1	1	1	1	Total number of times a student accessed the resource	0 or positive integer
Page	7	6	2	2	0		
Recourse	17	15	4	12	10		
Folder	2	0	17	12	0		
Url	0	0	1	1	0		
Assignments	8	9	8	8	9		
Submitted Assignments	3	9	8	8	9	Student Grades	[0, 10] decimal

It is worth noting that there were certain differences among the five courses (Tables 1 and 2). At first, they were offered by different departments (Physics and Chemical Engineering) and they had different format and content. Although courses C_2, C_4 and C_3, C_5 encompassed the same topic—that is, Physics and Chemistry, respectively—their content varied depending on the academic year of study. In addition, courses C_1, C_2, C_4 were theoretical (Physical Chemistry and Physics), while C_3, C_5 were laboratory courses (Analytical Chemistry Lab). Moreover, each course required the submission of a different number of assignments. Finally, it should be noted that different students attended these courses.

4.3. The Proposed Transfer Learning Approach

The present study intends to address the problem of transferring knowledge across different undergraduate courses. Hence, we employed a simple deep neural network architecture, comprised four layers: an input layer, two hidden dense layers and an output one. The input layer consists of input units corresponding to each one of the dataset input features (Table 3). The first hidden layer has 12 hidden units and the second one has eight. Both dense layers use the Relu activation function. Finally, the output layer consists of a single neuron employing the sigmoid activation function and the binary cross entropy loss function for predicting the output class (pass or fail).

The experimental procedure was divided into three distinct phases (Figure 2). In the first phase, we constructed all the unique pairs of courses that could be formed (ten pairs of courses in total). Each time, the related datasets were rebuilt to share a common set of features. For each pair of courses, we made use of the following notation:

$$\left\{C_i, C_j\right\} \; i, j \in \{1, 2, 3, 4, 5\}, \; i \neq j. \tag{1}$$

Table 3. Features of the paired datasets.

	$\{C_1,C_2\}$	$\{C_1,C_3\}$	$\{C_1,C_4\}$	$\{C_1,C_5\}$	$\{C_2,C_3\}$	$\{C_2,C_4\}$	$\{C_2,C_5\}$	$\{C_3,C_4\}$	$\{C_3,C_5\}$	$\{C_4,C_5\}$
Forum views	1	1	1	1	1	1	1	1	1	1
Page views	7	7	8	7	0	0	2	0	2	2
Recourse views	17	17	17	17	11	11	11	10	12	12
Folder views	2	2	0	2	12	0	12	13	12	12
URL views	0	0	0	1	0	0	1	1	1	1
Assignments views	13	8	9	8	9	9	9	9	8	9
Submitted Assignments	8	8	9	8	8	9	8	9	8	9
Total views	1	1	1	1	1	1	1	1	1	1
Total activity	1	1	1	1	1	1	1	1	1	1
Gender	1	1	1	1	1	1	1	1	1	1
Total Number of Features	51	46	47	47	44	33	47	46	47	49

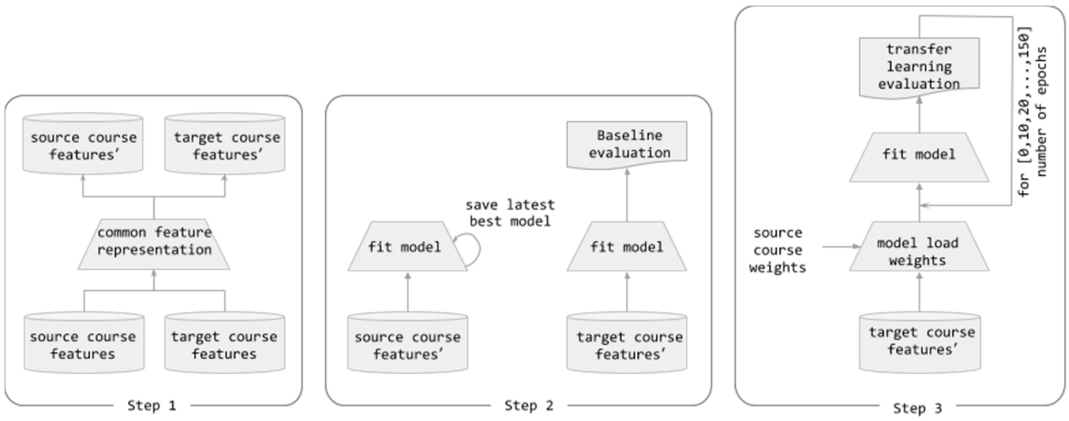

Figure 2. The three-step process of the proposed method.

In order to create a common set of features for each pair of courses, we matched features of the first course to related features of the second course one by one. Among the common features were the gender as well as the course total activity and course total views counters. Therefore, the first assignment of the first course was matched with the first assignment of the second course, the second assignment of the first course was matched with the second assignment of the second course and so forth, while the same procedure was followed for all the six types of resources. In cases where a matching feature was not found, a new feature was created, with zero values for each instance. For example, the C_1 course contained features related to seven page resources, whereas the C_2 course contained features related to six page resources (Table 2). Finally, the new $\{C_1,\ C_2\}$ pair of datasets contained seven features regarding the page resources, since a new empty feature was created and added in the C_2 course dataset, thus matching to the seventh feature of the C_1 course (Table 3).

The second phase refers to the training process of the two supporting deep networks. The first one was trained on the new source course C_i in order to extract its adjusted weights, while the second one was trained on the new target course C_j in order to calculate the baseline evaluation. In both cases, we calculated the accuracy metric, which corresponds to the percentage of correctly classified instances, while the models were trained for 150 epochs. In addition, the 10-fold cross validation resampling procedure was adopted for evaluating the overall performance of the deep network models.

The third phase was the most fundamental, since it implemented the transfer learning strategy. The deep model of the target course was fitted from scratch, but this time the network weights were initialized using the previously calculated weights from the source course (second phase). The pre-trained model was further tuned by running it each time for a certain number of epochs (hereinafter denoted as $C_{i,j}$): zero (i.e., the starting point), 10, 20, 30, 40, 50, 100 and 150. Algorithm 1 provides the pseudocode of the proposed transfer learning method. All the experiments were conducted using the Keras library in Python [31].

Algorithm 1: Transfer learning through paired course using deep neural networks.

```
Input: c1, c2, scores = [[] # c1 is the source course dataset, and c2 is the target
Output: scores # accuracy scores of the target course for every set of epochs
(c1', c2') ← commonRepresentation (c1, c2) # construct a common representation
model1 ← createAndCompileModel () # configure the deep learning process
weights ← fitModel (model1, c1', epochs=150) # train the model, and save its weights
model2 ← createAndCompileModel(weights) # load the weights of pre-trained model1
for each  e in [0, 10, 20, 30, 40, 50, 100, 150] do
    model2' ← fitModel (model2, c2', epochs= e) # further tune the pre-trained model2
    score ← evaluate (model2', c2', folds=10) # evaluate model2' using the accuracy metric
    add (score, scores) # save score for the output
end for each

# construct a common feature representation for the two courses
function commonRepresentation (dataset1, dataset2)
     dataset1'= [[], dataset2'= [[] # init empty datasets
     # match the features of dataset1 with the features of dataset2,
    # create new features when necessary
     for each t in ['forum', 'page', 'recourse', 'folder', 'url', 'assign views', 'assign'] do
        features1 ← getFeaturesOfType (dataset1, t) # get all features for this type
        features2 ← getFeaturesOfType (dataset2, t)
        size ← min (features1.size, features2.size)
        diff ← absoluteDifference (features1.size, features2.size)
        for i = 0 to size-1 do
           add (features1[i], dataset1')
           add (features2[i], dataset2')
        end for
        for j=0 to diff-1 do
           if f1 ← getFeatureAt(features1, features1.size + j) do # if f1 exists
              add (f1, dataset1')
              add (createEmptyFeature(), dataset2')
           else f2 ← getFeatureAt(features2, features2.size + j) # else f2 exists
              add (f2, dataset2')
              add (createEmptyFeature(), dataset1')
        end if
    end for
end for each
return dataset1', dataset2' # return the new datasets
```

5. Results

The averaged accuracy results (over the 10 folds) are presented in Table 4. For each pair, we conducted two experiments, using each course alternatively as the source course and the other one as the target course. Therefore, we evaluated 20 distinct combinations formed by the five courses. For each pair, we highlighted in bold the cases where the transfer model produced better results than the baseline. Overall, it is observed that the model $C_{i,j}$ benefits the predictions of the source course C_i, since the predictive performance of the transfer learning deep network is better than the baseline C_j.

Table 4. Averaged accuracy results.

	$\{C_1, C_2\}$		$\{C_1, C_3\}$		$\{C_1, C_4\}$		$\{C_1, C_5\}$		$\{C_2, C_3\}$	
	C_1	C_2	C_1	C_3	C_1	C_4	C_1	C_5	C_2	C_3
Baseline	0.7627	0.6094	0.7667	0.7047	0.7563	0.6333	0.7424	0.5591	0.6011	0.7144
Epochs	$C_{2,1}$	$C_{1,2}$	$C_{3,1}$	$C_{1,3}$	$C_{4,1}$	$C_{1,4}$	$C_{5,1}$	$C_{1,5}$	$C_{3,2}$	$C_{2,3}$
0	0.6106	**0.6172**	0.7524	**0.8227**	0.5675	**0.6371**	0.5889	**0.8644**	**0.6306**	0.6538
10	**0.7701**	**0.6414**	**0.7988**	**0.8382**	**0.8128**	**0.6387**	**0.8087**	**0.8561**	0.5975	**0.8234**
20	**0.7842**	**0.6299**	**0.7917**	**0.8537**	**0.8093**	0.5833	**0.8055**	**0.8553**	**0.6132**	**0.8382**
30	**0.7877**	**0.6132**	**0.8198**	**0.8537**	**0.8126**	0.6008	**0.8092**	**0.8644**	**0.6290**	**0.8394**
40	**0.7732**	**0.6234**	**0.8022**	**0.8608**	**0.8236**	0.6075	**0.8019**	**0.8553**	**0.6241**	**0.8453**
50	**0.7766**	0.6076	**0.8062**	**0.8465**	**0.8166**	0.6325	**0.7987**	**0.8386**	**0.6179**	**0.8537**
100	**0.7768**	0.5968	**0.7847**	**0.8465**	**0.8061**	**0.6762**	**0.7916**	**0.8114**	**0.6077**	**0.8394**
150	0.7552	**0.6185**	**0.7882**	**0.8620**	**0.7738**	0.6325	**0.7845**	**0.8371**	**0.6064**	**0.8472**
	$\{C_2, C_4\}$		$\{C_2, C_5\}$		$\{C_3, C_4\}$		$\{C_3, C_5\}$		$\{C_4, C_5\}$	
	C_2	C_4	C_2	C_5	C_3	C_4	C_3	C_5	C_4	C_5
Baseline	0.5650	0.6263	0.5498	0.6735	0.7096	0.6196	0.7715	0.7955	0.6811	0.6646
Epochs	$C_{4,2}$	$C_{2,4}$	$C_{5,2}$	$C_{2,5}$	$C_{4,3}$	$C_{3,4}$	$C_{5,3}$	$C_{3,5}$	$C_{5,4}$	$C_{4,5}$
0	**0.6207**	0.6008	0.4112	**0.8379**	0.4641	0.4988	**0.8156**	**0.8644**	0.5375	**0.6902**
10	**0.5941**	0.5817	**0.5916**	**0.8644**	**0.8310**	0.5892	**0.8156**	**0.8303**	0.5758	**0.7621**
20	**0.6154**	0.6133	**0.6031**	**0.8644**	**0.8394**	0.6067	**0.8239**	**0.8470**	0.5888	**0.8197**
30	**0.6120**	0.5946	**0.6188**	**0.8561**	**0.8465**	0.5563	**0.8322**	**0.8561**	0.6013	**0.8114**
40	**0.6074**	0.6192	**0.6130**	**0.8644**	**0.8406**	0.5883	**0.8310**	**0.8561**	0.6017	**0.8121**
50	**0.6330**	0.6258	**0.5938**	**0.8644**	**0.8549**	**0.6263**	**0.8251**	**0.8561**	0.6529	**0.8023**
100	**0.5885**	**0.6388**	**0.6272**	**0.8553**	**0.8322**	**0.6458**	**0.8329**	**0.8553**	0.6538	**0.7947**
150	**0.5947**	0.6133	**0.6105**	**0.7947**	**0.8251**	**0.6392**	**0.8299**	**0.8470**	0.6346	**0.8121**

A one-tailed, paired t-test (a=0.05) was conducted for verifying whether the improvement in the transfer model was statistically significant. Therefore, we compared the accuracy results obtained by the baseline deep network (using the target course dataset), with the results obtained by the transfer method, iteratively, for each number of epochs. Since the p-value is inferior or equal to 0.05, we conclude that the difference is significant in all cases except the starting point where the number of epochs equals zero (Table 5). Moreover, the p-value is gradually decreased as the number of epochs increase from every epoch from 10 to 100.

Table 5. The t-test results.

Epochs	p-Value
0	0.2449
10	0.0051
20	0.0023
30	0.0022
40	0.0013
50	0.0003
100	0.0002
150	0.0012

The analysis of the experimental results, in question-and-answer format, underlines the efficiency of the proposed method for transferring knowledge from one course to a related one.

1. Can the weights of a deep learning model trained on a specific course be used as the starting point for a model of another related course?

At the starting point for each transfer learning model (i.e., zero epochs) we used the weights estimated by the previously trained deep network models (on 150 epochs) instead of starting with

randomly initialized weights. For example, at the starting point of the $C_{4,2}$ transfer model, we used the weights estimated by the C_4 model.

Comparing the results of the pretrained weights without further tuning (i.e., zero epochs) to the baseline model, an improvement is noticed in half of the datasets (10 out of 20). The statistical results (t-test) confirm that the difference is not significant when the pre-trained model is not further tuned for the second dataset (target course C_j), since p-value=0.2449>a=0.05. However, the transfer model prevails in 16 out of 20 datasets when it is further tuned for only 10 epochs.

2. Will the pre-trained model reduce the training effort for the deep model of the second course?

Overall, the increase in the number of epochs improves the performance of the proposed transfer learning model. Moreover, the improvement is significant for every number of epochs, apart from the starting point, as statistically confirmed by the t-test results. It is worth noting that the transfer model prevails in 18 out of 20 datasets after 100 epochs, where the lowest p-value is 0.0002.

In addition, we can detect three cases of overfitting, since the accuracy ceases to improve after a certain number of iterations and begins to decrease. Particularly, this is observed in the cases where C_1 starts with C_2 weights, C_2 with C_1 weights and C_4 with C_1 weights. For instance, C_1 outperforms the baseline with an accuracy measure of 0.7768 after 100 epochs of retuning the preloaded weights of C_2. However, after 150 epochs the accuracy is decreased to 0.7552.

6. Discussion

An important finding to emerge in this study is that even a small amount of prior knowledge from a past course dataset could result in a fair measure of accuracy for predicting student performance in a related current course. This was verified by a plethora of experiments that have been carried out regarding twenty different pairs of five distinct one-semester courses, investigating the effectiveness of transfer learning in deep neural networks for the task of predicting at-risk students in higher education. In most cases, the transfer model obtained better accuracy than the baseline one. An improvement was noticed in half of the datasets (10 out of 20) using the pretrained weights from the source course (i.e., zero epochs). There was also a considerable accuracy improvement in most cases (16 out of 20) when the pre-trained model was further tuned for 10 to 40 epochs. Therefore, fine-tuning provides a substantial benefit over training with random initialization of the weights, thus leading to higher accuracy with fewer passes over the data. Overall, there was only one case where the transfer learning did not achieve better results ($C_{5,4}$). Hence, it is evident that it is not always feasible to transfer knowledge from one course to another one. In addition, it is worth noting that the type of course, laboratory or theoretical, does not seem to directly affect the predictive accuracy of the transfer learning model. This indicates that there is a slight uncertainty about the transferability level of a predictive model. The definition of what is a "transferable" model is where this ambiguity lies. A model trained on a set of courses is considered to be "transferable" if it achieves respectively fair results on a new, related course [10].

We believe this is yet another important attempt towards transfer knowledge in the educational field. Further, there are key issues to be considered such as measuring the degree of similarity between two courses (i.e., the number and form of learning activities), the type of attributes and the duration of the course. Finally, it is similarly important to build both simple and interpretable transferable models that could be easily applied by educators from one course to another [29]. Therefore, more studies are required on the current topic for establishing these results.

7. Conclusions

In the present study, an effort was made to propose a transfer learning method for the task of predicting student performance in undergraduate courses. The identification of failure-prone students could lead the academic staff developing learning strategies that aim to improve students' academic performance [32]. Transfer learning enables us to train a deep network using the dataset of a past course (source course) and reuse it as the starting point for a dataset of a new related course

(target course). Moreover, it is possible to further tune the repurposed model. Our findings proved that a fair performance was achieved in most cases, while the proposed method handily outperforms the baseline model.

Transfer learning offers many future research directions. Our results are encouraging and should be validated by larger samples of courses from different departments and programs. An interesting task is to apply a model for a specific task, such as the prediction of student's performance, for another related task, such as the prediction of student's dropout or for regression tasks (e.g., for predicting students' grades). In a future work we will also investigate the efficiency of transfer learning in imbalanced datasets obtained from several educational settings. If someone has only the target task, but also has the ability to choose a limited number of additional training data to collect, then active learning algorithms can be used to make choices that will improve the performance on the target task. These algorithms may also be combined into active transfer learning [33].

Author Contributions: Conceptualization, M.T. and S.K.; methodology, S.K.; software, M.T.; validation, G.K., S.K. and O.R.; formal analysis, M.T.; investigation, M.T.; resources, S.K.; data curation, G.K.; writing—original draft preparation, M.T.; writing—review and editing, G.K.; visualization, M.T.; supervision, S.K.; project administration, O.R.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pan, S.J.; Yang, Q. A survey on transfer learning. *Ieee Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
2. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
3. Brownlee, J. *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*; Machine Learning Mastery: Vermont Victoria, Australia, 2019.
4. Brownlee, J. *Deep Learning with Python: Develop Deep Learning Models on Theano and Tensorflow Using Keras*; Machine Learning Mastery: Vermont Victoria, Australia, 2016.
5. Ng, A. Nuts and bolts of building AI applications using Deep Learning. Nips Keynote Talk. In Proceedings of the Thirtieth Conference on Neural Information Processing Systems. 2016 NIPS'16, Barcelona, Spain, 5–10 December 2016.
6. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
7. Liz-Domínguez, M.; Caeiro-Rodríguez, M.; Llamas-Nistal, M.; Mikic-Fonte, F.A. Systematic literature review of predictive analysis tools in higher education. *Appl. Sci.* **2019**, *9*, 5569. [CrossRef]
8. Boyer, S.; Veeramachaneni, K. Transfer learning for predictive models in massive open online courses. In *International Conference on Artificial Intelligence in Education*; Springer: Berlin/Heidelberg, Germany, 2015.
9. Ding, M.; Wang, Y.; Hemberg, E.; O'Reilly, U.-M. Transfer Learning using Representation Learning in Massive Open Online Courses. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, Tempe, AZ, USA, 4–8 March 2019.
10. Boyer, S.A. Transfer Learning for Predictive Models in MOOCs. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2016.
11. Guo, B.; Zhang, R.; Xu, G.; Shi, C.; Yang, L. Predicting students performance in educational data mining. In Proceedings of the 2015 International Symposium on Educational Technology (ISET), Wuhan, China, 27–29 July 2015.
12. Okubo, F.; Yamashita, T.; Shimada, A.; Ogata, H. A neural network approach for students' performance prediction. In Proceedings of the Seventh International Learning Analytics & Knowledge Conference, Vancouver, BC, Canada, 13–17 March 2017.
13. Kim, B.-H.; Vizitei, E.; Ganapathi, V. GritNet: Student performance prediction with deep learning. *arXiv* **2018**, arXiv:1804.07405.

14. Kostopoulos, G.; Tsiakmaki, M.; Kotsiantis, S.; Ragos, O. Deep Dense Neural Network for Early Predic-tion of Failure-Prone Students. In *Machine Learning Paradigms-Advances in Theory and Applications of Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2019.
15. Wang, W.; Yu, H.; Miao, C. Deep model for dropout prediction in MOOCs. In Proceedings of the ACM International Conference Proceeding Series, Beijing, China, 6–9 July 2017.
16. Whitehill, J.; Mohan, K.; Seaton, D.; Rosen, Y.; Tingley, D. Delving Deeper into MOOC Student Dropout Prediction. *arXiv* **2017**, arXiv:1702.06404.
17. Xing, W.; Du, D. Dropout prediction in MOOCs: Using deep learning for personalized intervention. *J. Educ. Comput. Res.* **2018**, *57*, 547–570. [CrossRef]
18. Bosch, N.; Paquette, L. Unsupervised Deep Autoencoders for Feature Extraction with Educational Data. In Proceedings of the Deep Learning with Educational Data Workshop at the 10th International Conference on Educational Data Mining, Wuhan, Hubei, 25–28 June 2017.
19. Ruder, S.; Peters, M.E.; Swayamdipta, S.; Wolf, T. Transfer learning in natural language processing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, Minneapolis, MN, USA, 2 June 2019.
20. Weiss, K.; Khoshgoftaar, T.M.; Wang, D.D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1345–1459. [CrossRef]
21. Arnold, A.; Nallapati, R.; Cohen, W.W. A Comparative Study of Methods for Transductive Transfer Learning. In Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, Omaha, NE, USA, 28–31 October 2007.
22. Romero, C.; Ventura, S. Educational data mining and learning analytics: An updated survey. *Wiley Interdiscip. Rev.* **2020**, e1355. [CrossRef]
23. Moreno-Marcos, P.M.; Alario-Hoyos, C.; Muñoz-Merino, P.J.; Kloos, C.D. Prediction in MOOCs: A review and future research directions. *IEEE Trans. Learn. Technol.* **2018**, *12*, 384–401. [CrossRef]
24. Costa, E.B.; Fonseca, B.; Santana, M.A.; de Araújo, F.F.; Rego, J. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Comput. Hum. Behav.* **2017**, *73*, 247–256. [CrossRef]
25. Vitiello, M.; Walk, S.; Chang, V.; Hernandez, R.; Helic, D.; Guetl, C. MOOC dropouts: A multi-system classifier. In Proceedings of the European Conference on Technology Enhanced Learning, Tallinn, Estonia, 12–15 September 2017.
26. Hunt, X.J.; Kabul, I.K.; Silva, J. Transfer Learning for Education Data. In Proceedings of the KDD Workshop, Halifax, NS, Canada, 13–17 August 2017.
27. Tri, P.T.; Chau, V.T.N.; Phung, N.H. Combining transfer learning and case-based reasoning for an educational decision making support model. In Proceedings of the Multi-disciplinary Trends in Artificial Intelligence: 11th International Workshop, MIWAI 2017, Gadong, Brunei, 20–22 November 2017.
28. Zeng, Z.; Chaturvedi, S.; Bhat, S.; Roth, D. DiAd: Domain adaptation for learning at scale. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, Tempe, Arizona, 4–8 March 2019.
29. López-Zambrano, J.; Lara, J.A.; Romero, C. Towards Portability of Models for Predicting Students' Final Performance in University Courses Starting from Moodle Logs. *Appl. Sci.* **2020**, *10*, 354. [CrossRef]
30. Tsiakmaki, M.; Kostopoulos, G.; Kotsiantis, S.; Ragos, O. Implementing AutoML in Educational Data Mining for Prediction Tasks. *Appl. Sci.* **2019**, *10*, 90. [CrossRef]
31. Chollet, F. Keras. Available online: https://keras.io (accessed on 1 January 2020).
32. Romero, C.; Ventura, S. Data mining in education. *Wiley Interdiscip. Rev.* **2013**, *3*, 12–27. [CrossRef]
33. Wang, X.; Huang, T.-K.; Schneider, J. Active transfer learning under model shift. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.