Diving into the Deep End of Machine Learning: Using Keras for Predicting Education Outcomes with Tabular Data

Elena Volpi

March 14, 2023

Background and Motivation

Deep Learning

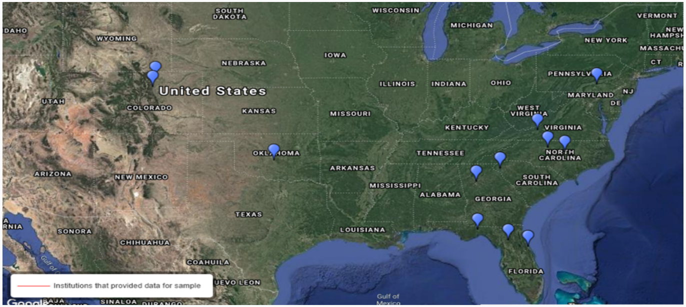
- New wave of deep learning popularity
- Advancements: abundance of data from internet, much more powerful computer processing chips
- Examples: Natural language processing, image recognition, handwriting recognition
- Classification and Regression
- Highly sought after job skill

Project Goals: Gain experience with deep learning and TensorFlow, apply deep learning on large and complex data set for predicting STEM graduates

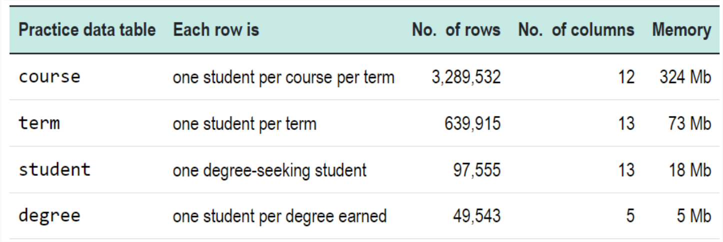
Outline

Multiple Institution Database for Investigating Engineering Longitudinal Development

- Created in 2004
- Contains data from 2.2 million undergraduates from 20 institutions across the US from 1988-2018.
- Public stratified sample has 97,640 undergraduates from 12 institutions from same years.
- All institutions must offer an engineering degree



MIDFIELD cont.



Practice data table	Each row is	No. of rows	No. of columns	Memory
`course`	one student per course per term	3,289,532	12	324 Mb
`term`	one student per term	639,915	13	73 Mb
`student`	one degree-seeking student	97,555	13	18 Mb
`degree`	one student per degree earned	49,543	5	5 Mb

MIFIELD is composed of four tables:

- student: entering demographic information about the student.
- term: contains information on standing, GPA, credit hours, and any changes in major.
- course: contains every course the student has taken (grade, class number, section, level)
- degree: if a student obtained a degree, degree shows specific degree and term completed

MIDFIELD cont.

Important variables:

- MCID: Unique ID number to identify each student.
    - serves as the primary key, meaning it connects information for each student across all four tables.
    - Exp: MID25783178
- CIP6:
    - Classification of Instructional Programs (CIP) is a standardized code for majors across the US.
    - CIP6 used to identify declared majors in the dataset.
    - First 2 numbers are field of study, final four digits are more specified degrees.
    - exp. All CIP6 that start with 14 are engineering, 143501 is Industrial Engineering.
    - Undeclared majors are 999999

Deep Learning

Tensors

The data structure of deep learning are tensors: multidimensional arrays with arbitrary number of dimensions.

Attributes:

- Rank: number of axes
    - A 3D tensor has rank 3 and three axes.

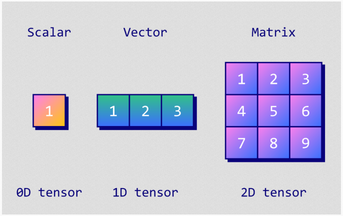


photo: https://mlabonne.github.io/blog/what-is-a-tensor/

Tensors

The data structure of deep learning are tensors: multidimensional arrays with arbitrary number of dimensions.

Attributes:

- Rank: number of axes
- Shape : dimensions in each axis.
    - A timeseries tensor has shape (samples, timesteps, features)

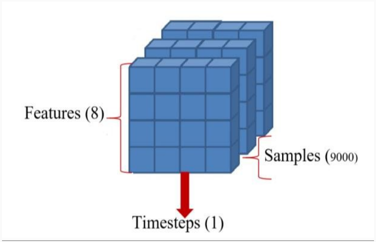


photo:https://www.researchgate.net/figure/D-tensor-for-US-Oil-and-Gas-Production-Time-series-Dataset-J-o-u-r-n-a-l-P-r-e-p-r-o-o-f_fig4_346445827

Tensors

The data structure of deep learning are tensors: multidimensional arrays with arbitrary number of dimensions.

Attributes:

- Rank: number of axes
- Shape : dimensions in each axis.
- Type: any numeric type
    - Integers or real numbers

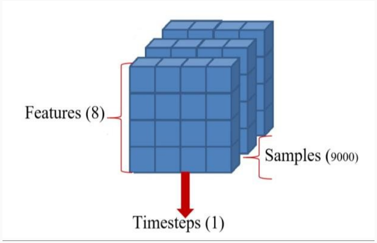


photo:https://www.researchgate.net/figure/D-tensor-for-US-Oil-and-Gas-Production-Time-series-Dataset-J-o-u-r-n-a-l-P-r-e-p-r-o-o-f_fig4_346445827

Deep Learning Terminology

Given student data, the deep learning model will attempt to classify each student as a STEM grad or not a STEM grad.

- Inputs are explanatory variable/covariates
- True outcomes are called labels or targets
- Deep learning maps inputs to targets variables through tensor operations called layers.
- Weights: describes how inputs are transformed and sent to proceeding layers

Deep Process Overview

We'll first give a broad overview of the deep learning process:

1. Break data into training and testing sets with corresponding training and testing labels.
2. Input training set and obtain predictions.
3. Compute the loss,the difference between our true outcomes and predictions.
4. Adjust the weights of our model to slightly reduce the loss and repeat steps.

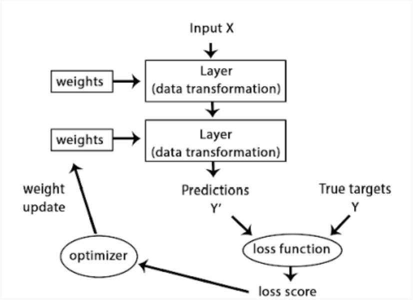


Figure 1: Training Loop

photo: Chollet, Francois, and J. J. Allaire. Deep Learning with R. Manning pg. 11, 2018

Layers

- Data transformations that return new data representations
- Incrementally build more complex representations
- For sequential models (our model type), layers are stacked where all data goes through consecutive layers.

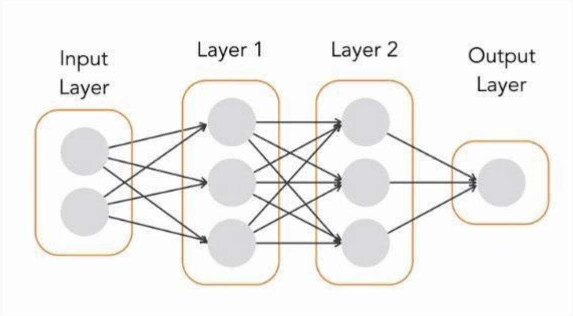


photo: https://www.researchgate.net/figure/Fig-2-A-sequential-neural-model-Keras-Sequential-Api-nd-Activation-Functions-The_fig2_350567223/download

Layers

- We'll use two types of layers, densely connected layers and dropout layers.
    - **Densely connected layers/ Dense Layers** : each unit/node in each layer connects with each node/layer in the following layer
    - **Dropout layers**: set a percentage of features to zero to prevent over fitting.
- For all layers, we will specify layer type, units, activation functions and for the input layer, shape = # columns in training data.

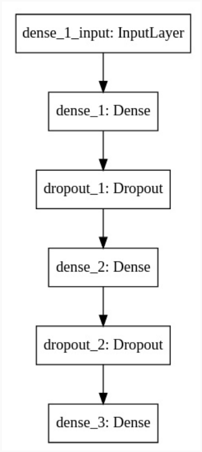


photo: https://www.kdnuggets.com/2018/06/basic-keras-neural-network-sequential-model.html

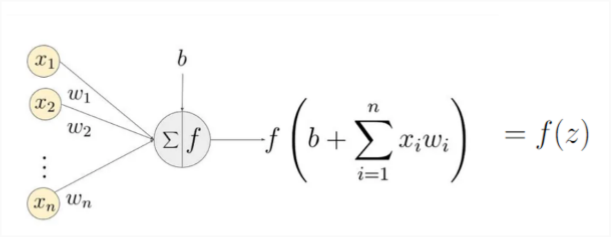
Units

- Units, or hidden units: the input dimension of the weight matrix W.
- High units give more complex representations.
    - Cons : higher computation expense, longer computation time, overfitting

Weight Matrix and Activation Functions

The weight matrix, W is an $n \times m$ matrix with elements w_{ij}.

- the parameter within a neural network that transforms input data within the network's hidden layers.
- `z = ` $(\mathrm{dot}(W, X) + b)$
    - W is our weight matrix, X is our input, b is bias,
    - dot is the dot product between input tensor and W



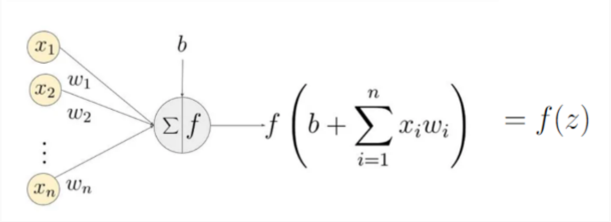
* for densely connected layers and linear activation function, f

photo: https://learnopencv.com/understanding-activation-functions-in-deep-learning/

Activation Functions

The activation function, f, decides what info is "important" and should be sent to the next node.

- also known as transfer function
- transforms the aggregate z (weighted input) and outputs to the next node
- Can be linear* or non-linear
- Must be differentiable



* for densely connected layers and linear activation function, f
photo: https://learnopencv.com/understanding-activation-functions-in-deep-learning/

Activation Functions: Rectified Linear Unit

- Rectified Linear Unit (relu) transform the inputs so that negative values are zero and positive values are positive real numbers
- $\texttt{relu} = \texttt{max}(x, 0)$
- piece-wise linear

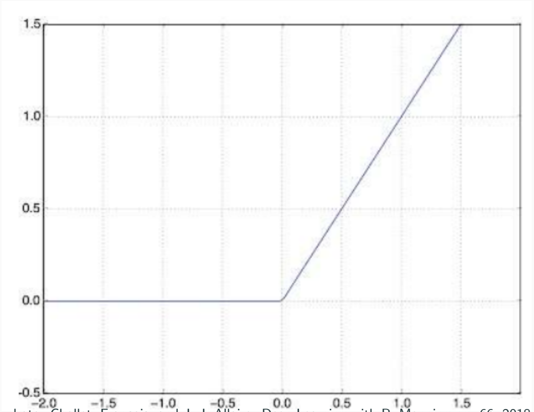


photo: Chollet, Francois, and J. J. Allaire. Deep Learning with R. Manning pg. 66, 2018

Activation Functions: Sigmoid

- Sigmoid : $y = \frac{1}{1+e^{-x}}$
- Flattens input values into the [0,1] interval
- Used in output layer to output a probability

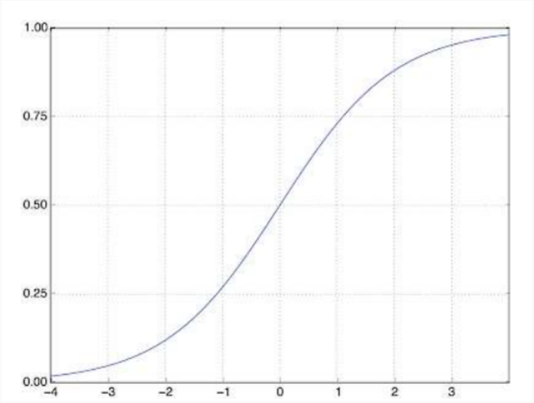
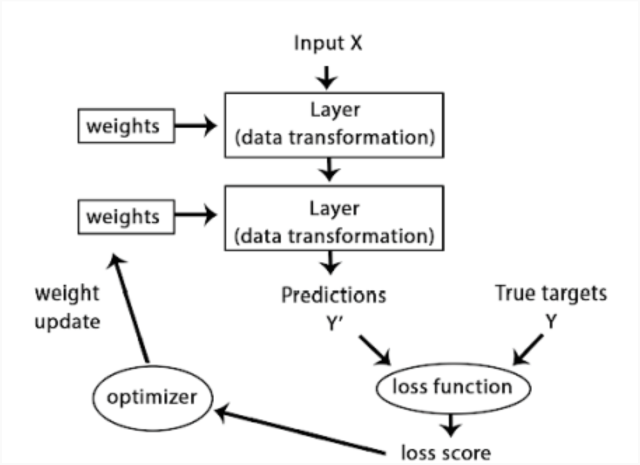


photo: Chollet, Francois, and J. J. Allaire. Deep Learning with R. Manning pg. 67, 2018


Input X
weights
Layer
(data transformation)
weights
Layer
(data transformation)
weight
update
Predictions
Y'
True targets
Y
optimizer
loss function
loss score

Loss function

- Loss function: defines how we compute the distance from the prediction to true target value and is what we strive to minimize
- Binary classification: loss function we use is log-loss or binary crossentropy

$$L_{BCE} = \frac{1}{M}\sum_{i=1}^{M} y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i)), \tag{1}$$

- y is our labels
- $p(y_i)$ is the probability that this observations belongs to the "yes" group
- M is the number of observations

Optimizer and Backpropagation

- After the loss function calculates distance score, it passes it to the optimizer.
- The Optimizer determines how we should adjust the weights using backpropagation.
- Backpropagation uses gradient descent to adjust W and b in order to find weights that minimize loss.
    - The gradient is the derivative of a tensor operation (our layers)

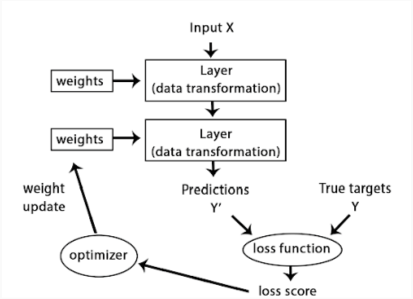


Figure 1: Forward Pass through the network

Backpropagation

- Using the gradient or mean of the gradient of a batch of samples, we propagate the model from input layer to output layer, iteratively adjusting weights.
- Backwards Pass: $W = W - (step * gradient)$
- We used the optimizer `adam`, an extension of stochastic gradient descent and makes use of the average of uncentered variance.

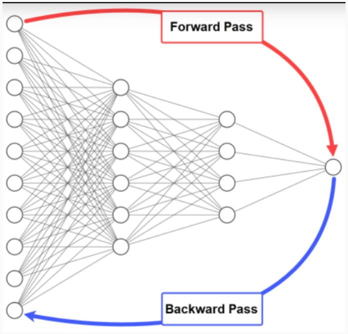


photo: https://neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide

Metrics

- Metrics: monitor the performance of training and testing
- Choice depends on whether classification or regression item
- Our choice:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- Recall was chosen since we have unbalanced data and it measures how well the model is doing predicting the minority class
- True positives will be students correctly identified as STEM grads
- False negatives will be students that are STEM grads but were predicted not to be.

- **Keras metrics$_r$ecall()doesmajorityclass*

Model Fitting

In the model fitting stage we must define the number of epochs, batch size, and validation split.

- Epoch : one training cycle for the neural network
    - Too few epochs: not enough weight adjustments for good predictions, unstable
    - Too many epochs: overfitting, higher computation time, expensive
- Batch size : number of samples in one backwards pass
    - Small batch size: Long computation time, higher expense
    - Large batch size: Higher memory usage
- Validation split : percentage of data withhold from the training data as to evaluate the model's performance without exposing it to the testing data

Data Wrangling

Data Wrangling

Deep learning restrictions:

- No NA's can be in the data
    - Keras will not tell you if you have NA's
    - We experienced non-relevant errors, loss of NA and computer crashing/high RAM usage instead
- Data must be numeric
    - Encode characters
    - Standardize nominal variables: results in the best loss
- Need to get all data on the student level, "widening" the data so each row represents one distinct student
- Make variable decisions as to not overfit or give highly correlated explanatory variables

Data Wrangling: Student

After wrangling, the student dataframe has 10 variables, 97,639 rows.

- **mcid** : Unique student id, the primary key
- Binary encoded variables:
    - **sex**: 1 - Male, 0 - Female
    - **us_citizen**
    - **transfer**
- **race**: 1-Asian, 2-Black, 3-Hispanic/Latinx, 4-International, 5-Native American, 6-Other/Unknown, and 7-White
- Real-valued numeric rescaled to be between [0,1]
    - **hours_transfer**
    - **age**
    - **sat_math**
    - **sat_verbal**
    - **act_comp**

Data Wrangling: Term

- Our wrangled term table is of dimension 95943x10
- **mcid**
- **initial_cip6**: Student declared major in first term
- **most_prevelant_cip6**: The major code the student declared for the most terms
- **final_cip6*** : The major code in the student's last major
- Real-valued numeric rescaled to be between [0,1]
    - **major_changes**: Number of major changes
    - **highest_level**: Highest grade level recorded (i.e. 2-sophomore, 3-junior)
    - **NumGoodTerms**: Number of terms student was in good academic standing
    - **NumProbationTerms**: Number of terms student was in academic probation
    - **avg_hours_per_term**: mean hours student completed per term
    - **final_gpa**: cumulative gpa in student's last term

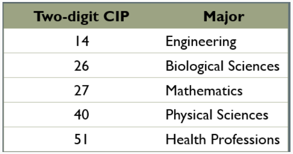
Data Wrangling: Course

Course has to be flatted from 3.5 million rows to 96644 rows and 3640 variables. When we restrict to only freshmen and sophomore level courses, we have 1603 variables.

- **mcid**
- **num_W**: number of withdrawals the student has
- **num_I**: number of incompletes the student has
- **num_WI**: number of combined withdrawals and incompletes a student has.
- The rest of the variables follow the form Institution (A-L) course abbrev lower/upper division: exp Institution A MTH lower
  - The variable entries are the product of grade in 4.0 GPA scale (exp. B+ is 3.3) and credit hours. We rescaled all columns.

Data Wrangling: Identifying STEM majors and grads

- The Department of Homeland Security has determined a list of CIP major codes that qualify as a STEM major.
- Label/Target: **is_stem_grad** : 1 student obtained a STEM degree, 0 if not.
    - Out of the 47,499 students who obtained a degree, 11,864 received a STEM degree.
- Also created feature, **initial_cip6_stem**, binary indicator of whether first major declared was STEM



Two-digit CIP	Major
14	Engineering
26	Biological Sciences
27	Mathematics
40	Physical Sciences
51	Health Professions

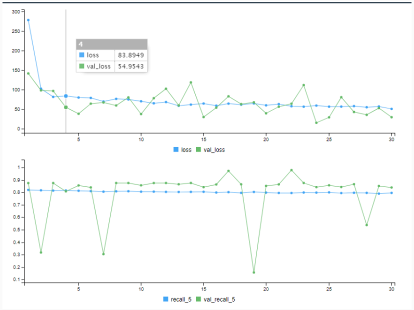
Data Wrangling

- Join the data using left joins and replace created NA's
    - replacing NA's with -1 if 0's were meaningful in the column and 0's if not.
- Created two dataframes:
    - student + terms + initial_cip6_stem: 95,942 students with 19 features
    - lower_div_course: 96,644 students and 1603 variables

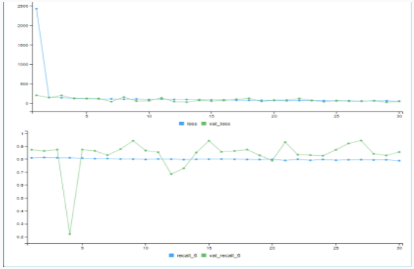
Models

Model 1 : Student_terms

- Split data into training and testing, 70% for training
- 30 epochs, 32 batch size, 0.3 validation
- Tried 20 models: different number of layers, units, and with/without dropout layers
- Faced two main issues with model fitting: high variability and accuracy/recall

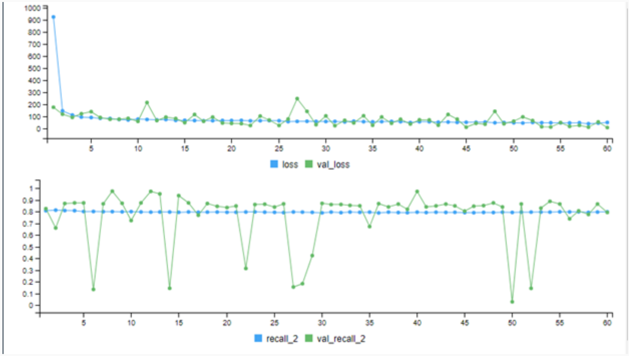


High variability: loss fails to converge, recall also unstable



Although we have nice flat loss, recall is 0, meaning we predicted 0 STEM majors.

Model 1 : Student_terms



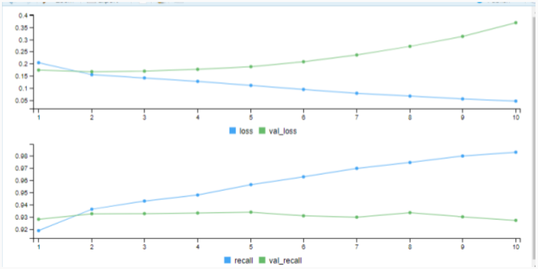
- Final model: Two dense layers with 16 units and 2 units.
- Recall (hand calculated for minority class): 0.46

Table 1

	Actual Non-STEM	Actual STEM
Predict Non-STEM	22012	1883
Predict STEM	321	1607

Model 2: lower_div_courses

- Same validation split and batch size. Fewer epochs because model is more stable due to much more data.



- Initial model was layer1: dense, units = 128, layer2: dense , units = 64, layer3: dense, units = 2.
- We can see overfitting as val_loss is curving upwards
- We add in dropout layers and experiment with different rates, number of layers, amount of units

Model 2: lower_div_courses

Final model

- Layer 1: dense, 64 units
- Layer 2: droupout, rate = 0.5
- Layer 3: dense, units = 32
- Layer 4 /Output layer: dense, units = 2
- Recall (hand calculated for minority class) = 0.72

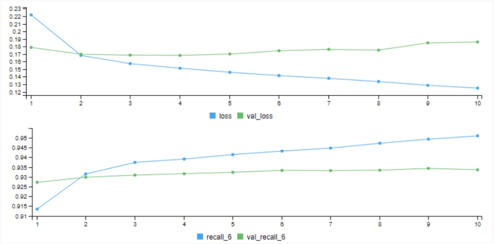


Table 2

	Actual Non-STEM	Actual STEM
Predict Non-STEM	24720	1232
Predict STEM	684	2358

Comparison to logistic regression

```
glm(y_train_labels ~ ., data= x_train_tbl, family =
binomial)
```

- student+terms+ initial_cip6 glm recall is 0.4971347
- Deep learning student+terms_initial_cip6 recall was 0.46

Table 3: GLM student_terms_cip6

	Actual Non-STEM	Actual STEM
Predict Non-STEM	24435	1755
Predict STEM	858	1735

Table 1: Model 1

	Actual Non-STEM	Actual STEM
Predict Non-STEM	22012	1883
Predict STEM	321	1607

Comparison to logistic regression

We'll now look at comparing the deep learning lower_division_courses model, model 2, to a glm model on the same data.

- **GLM**: recall= 0.6426184
- **Model 2**: recall = 0.72

Table 4: GLM lower_div_courses

Group	Actual Non-STEM	Actual STEM
Predict Non-STEM	24687	1283
Predict STEM	717	2307

Table 2: Model 2

Group	Actual Non-STEM	Actual STEM
Predict Non-STEM	24720	1232
Predict STEM	684	2358

Was it worth it?

- Performance between glm and deep learning was comparable
- For lower_div_courses, deep learning was 1.5 hours faster
- However, deep learning required significant data wrangling

Was it worth it?

- Walking away with a fairly good understanding of deep learning/ TensorFlow/ Keras
- Improved Coding Skills
- Enjoyed deep learning and project experience

Thank you! Questions?

References

Baheti, Pragati. “Activation Functions in Neural Networks [12 Types amp; Use Cases].” V7, 2 Mar. 2023, https://www.v7labs.com/blog/neural-networks-activation-functions.

Brownlee, Jason. “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.” MachineLearningMastery.com, 12 Jan. 2021, https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

Chollet, Francois, and J. J. Allaire. Deep Learning with R. Manning, 2018

CIP User Site, National Center for Educational Statistics, 2020, nces.ed.gov/ipeds/cipcode/search.aspx?y=56.

Dancho (2018, Jan. 11). Posit AI Blog: Deep Learning With Keras To Predict Customer Churn. Retrieved from https://blogs.rstudio.com/tensorflow/posts/2018-01-11-keras-customer-churn/

“Eligible CIP Codes for the STEM OPT Extension: Study in the States.” Eligible CIP Codes for the STEM OPT Extension — Study in the States, Department of Homeland Secuity, 2016, studyinthestates.dhs.gov/stem-opt-hub/eligible-cip-codes-for-the-stem-opt-extension.

Layton R, Long R, Ohland M, Orr M, Lord S (2022). _midfieldr: Tools and Methods for Working with MIDFIELD Data in 'R'_. R package version 1.0.0.9030, https://midfieldr.github.io/midfieldr

Layton R, Long R, Ohland M (2021). _midfielddata: Sample of MIDFIELD Student Unit Record Data_. R package version 0.1.3, https://github.com/MIDFIELDR/midfielddata