Deep Learning vs. AI Course

Elsa Mattson, Johnny Shapiro, Hui Wang, Tyler Williamson

What is Deep Learning and Neural Networks?

- A neural network takes an input vector of p variables and builds a nonlinear function f(X) to predict the response Y.
- The cornerstone of deep learning is the neural network
- The structure of neural networks distinguishes them from other nonlinear prediction models.

More on Deep Learning

- The number of p predictors (features) make up the units in the input layer of the NN
- Neural networks can have hidden layers, which are additional computations of nonlinear transformations of linear combinations of the units in the input layer.
    - These hidden layer functions are not fixed in advance, but learned during the training of the network
- Activation functions must be nonlinear to specify model’s nonlinearity
- The sum of two nonlinear transformations of linear functions gives an interaction
- Fitting a neural network requires estimating the unknown parameters in the model, usually done with squared-error loss

Why and When to Use Deep Learning

Deep learning should be used for larger and more complex datasets. The higher the volume, the more valuable deep learning will be. Deep learning is not as effective in situations where datasets are simple, budgets are low, and the problems at hand can be easily and quickly solved.

Deep learning may also require specialized hardware and high computing power.

Examples of deep learning include assigning classes to images or enhancing visuals.

Regressor vs Classifier

The two most common use cases for neural networks are regressors and classifiers, both of which are used to make predictions of a dependent variable based on predictor features of an object

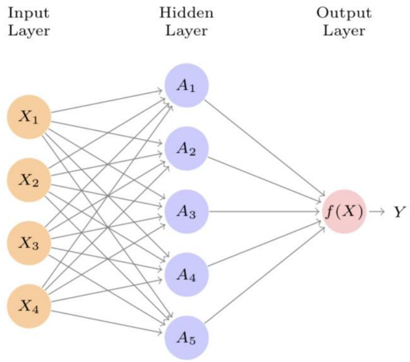
- Regressor - performs a regression on the dataset and predicts a value for the specified dependent variable. The metric used to measure accuracy is Mean Squared Error, and we want the lowest possible value we can achieve.
- Classifier - performs a classification on the dataset, creating bins for each possible value for the dependent variable and using one-hot encoding to assign a value of 1 to the bin the neural network predicts the observation belongs to, and a 0 to all the other bins.

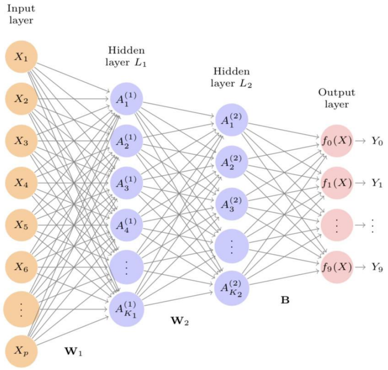
Regressor neural networks are used to predict continuous values like price, age, salary, etc. while Classifier neural networks are used to predict categroical values, such as a score on a scale of 1-10, sex, true/false, etc.

Single Layer vs Multi Layer

A neural network takes an input vector of p variables and builds a nonlinear function f(X) to predict the response Y. The number of p predictors (features) make up the units in the input layer of the NN. Neural networks have hidden layers that perform nonlinear transformations on the input vector before it goes into the output layer.

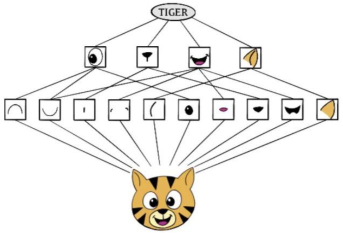
A neural network can have one or multiple hidden layers, called that because the values they produce aren't visible. If a NN has more layers, learning becomes much easier for it. Each layer randomizes the weights it assigns to each feature, which allows it to learn more and produce a more accurate answer.





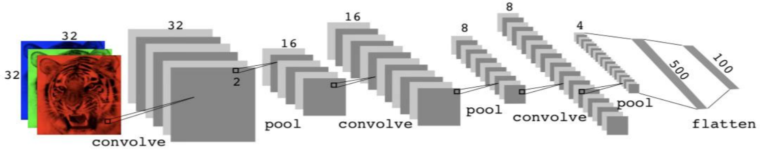
Convolutional Neural Network

Convolutional neural networks are an upgrade to the standard single and multi hidden layer neural networks of the past. They were created in an attempt to replicate human decision making processes to allow for classification operations on more variable and detailed datasets, such as a dataset of over 100 different species of animals. The network identifies **low-level** features first (i.e. edges, lines, colors), followed by **high-level** features (parts of eyes, ears).



Convolutional NNs are made up of two types of layers

- Convolution layers run the data through filters to create convolved images, assigning values to combinations within the convolved image that more closely resemble the original image.
    - The filters used in each convolution layer are learned for each specific classification task.
- Pooling layers condense large images into smaller summary images. These pooled images are sent into the next convolution layer.



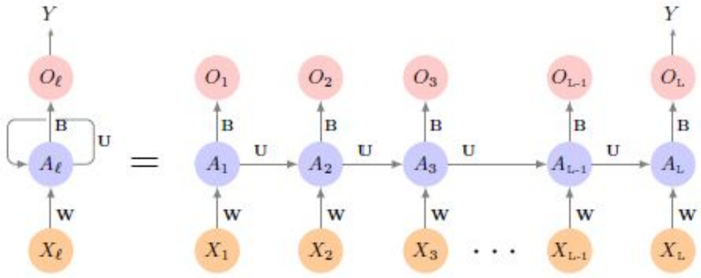
Document Classification

- Important applications in industry and science
  - Predicting attributes of documents, such as articles, reviews, or medical journals
- How do we ***featurize*** documents and define our sets of predictors?
  - Score each document for the presence of words in a language dictionary using **one-hot encoding**
  - Runs into the issue of having vectors of length = **M** ,where **M** = Total number of words in language dictionary
  - Must make a training ***corpus*** of the most frequently occurring words
- **Bag-of-words model**
  - ***Sparse Matrix*** - eg. we have a training feature matrix with dimension 25,000 x 10,000
  - Fit a lasso logistic regression using *glmnet* **AND** a two-class NN with hidden layers
  - Ignores context of the words
- **Bag-of-n-grams model**
  - A bag of 2-grams records co-occurrences of every distinct pair of words
  - “Blissfully long” versus “Blissfully short”

Recurrent Neural Network

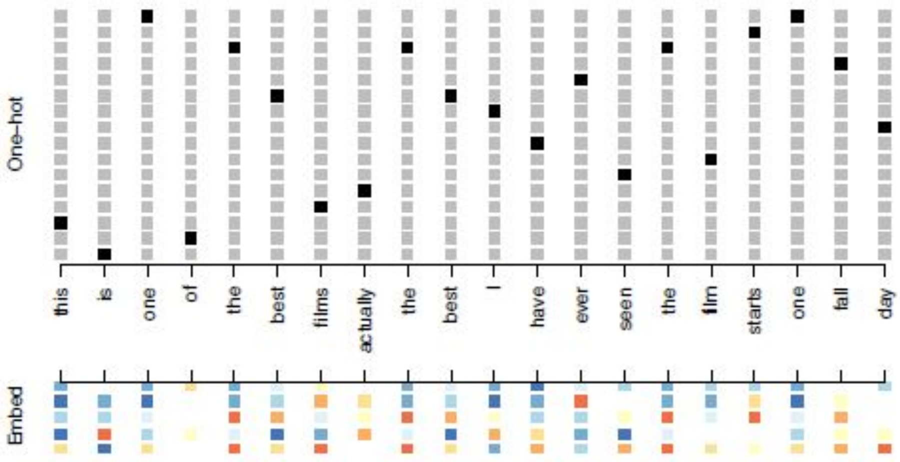
- For data sources that are **sequential** in nature, where data captures theme, tone, or sentiment and can be used for sentiment analysis, topic classification, and language translation
- Applications in weather and financial forecasting, sound recordings, and handwriting analysis
- We use a ***Corpus*** of documents, each with **L** words, where **X** represents a word
- Similar to how CNN's accommodate spatial features, RNN's accommodate sequences

$$X = \{X_1, X_2, \ldots, X_L\}$$



Recurrent Neural Network - IMDb Example

- Making predictions on the label of an entire document (*positive* or *negative* movie review)
- Dimensionality issue → each word is represented by one-hot encoded vectors of length 10,000
    - How do we fix this?
- Instead of one-hot encoding, represent each word with a set of m real numbers, known as **embedding** features in an m-dimensional space, known as the **embedding layer** ("E")



Weight Freezing
Word2vec or GloVe

Time Series Forecasting

- Applications in weather and financial forecasting
- Day-to-day observations are not independent of each other, but rather similar
- Autocorrelation
  - The correlation coefficient at lag **L** between observations that are a lag of **L** days apart
- Once again, we construct our input **X** sequences with a predefined length and target **Y**
- Here, each input **X** is 3-vectors consisting of trading volume, Dow Jones Return, and log volatility
  - Similar to Autoregression models we learned in Stochastic Modeling!

$$X_1 = \begin{pmatrix} v_{t-L} \\ r_{t-L} \\ z_{t-L} \end{pmatrix}, X_2 = \begin{pmatrix} v_{t-L+1} \\ r_{t-L+1} \\ z_{t-L+1} \end{pmatrix}, \cdots, X_L = \begin{pmatrix} v_{t-1} \\ r_{t-1} \\ z_{t-1} \end{pmatrix}, \text{ and } Y = v_t.$$

Fitting a Neural Network

- Optimization problems with multiple solutions
- Use gradient descent to reach a local minimum or maximum
- Techniques such as dropout learning and network tuning seen in AI
- Regularization
    - Penalties imposed on the model parameters, usually by lasso or ridge
    - Limiting epochs such to avoid overfitting

R Tutorial