Lecture 06
Deep Learning: Advanced

February 27, 2025

Data Science Using Python

Jaeung Sim

Assistant Professor

School of Business, University of Connecticut

Contents

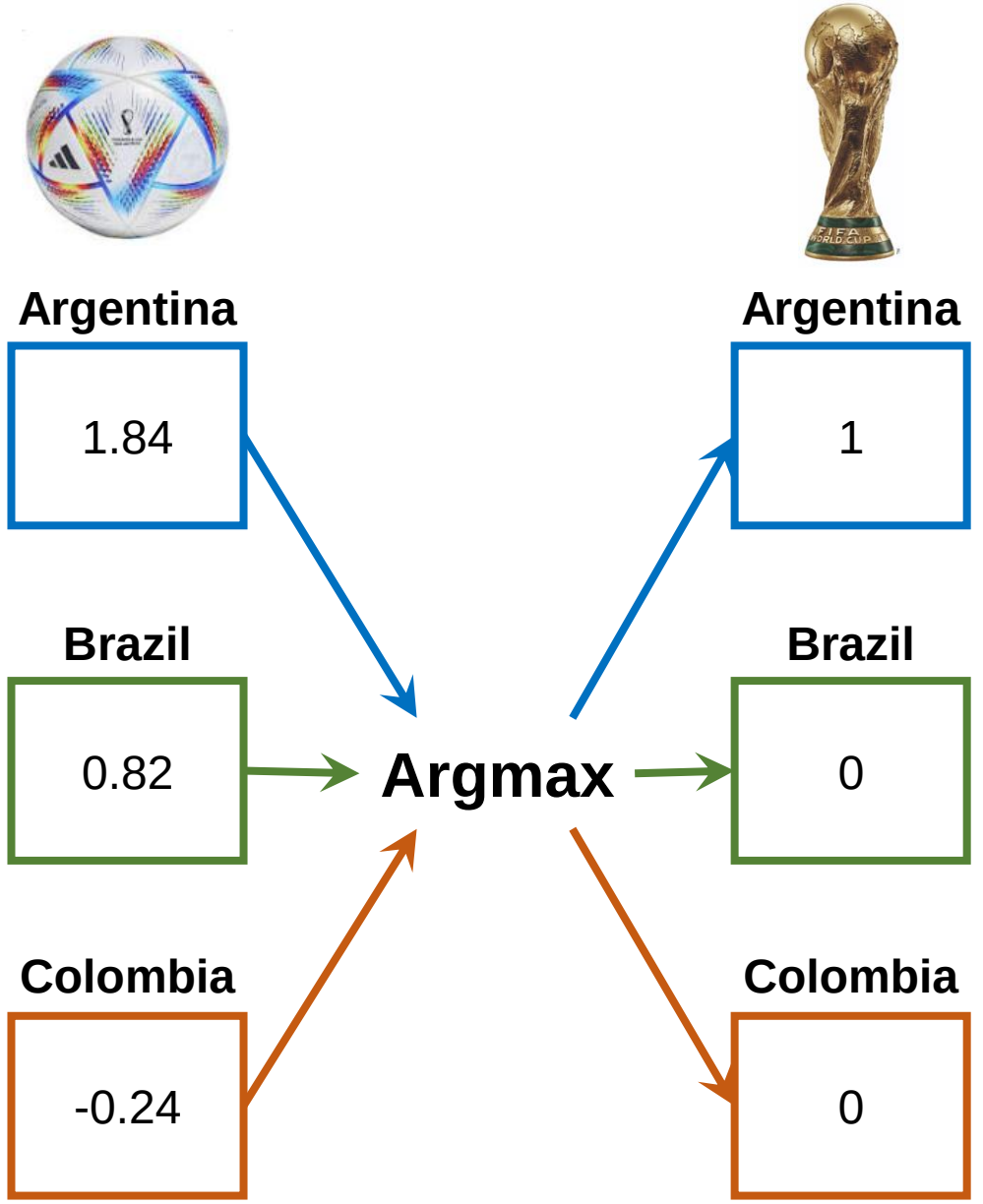
- **Deep Learning Details**
  - Argmax, Softmax
  - Cross Entropy
  - Epochs
  - Batch Size
  - Dropout
  - Summary of Hyperparameters
- **Notice for Upcoming Weeks**

Deep Learning Details

Details of Hyperparameters

Argmax, Softmax

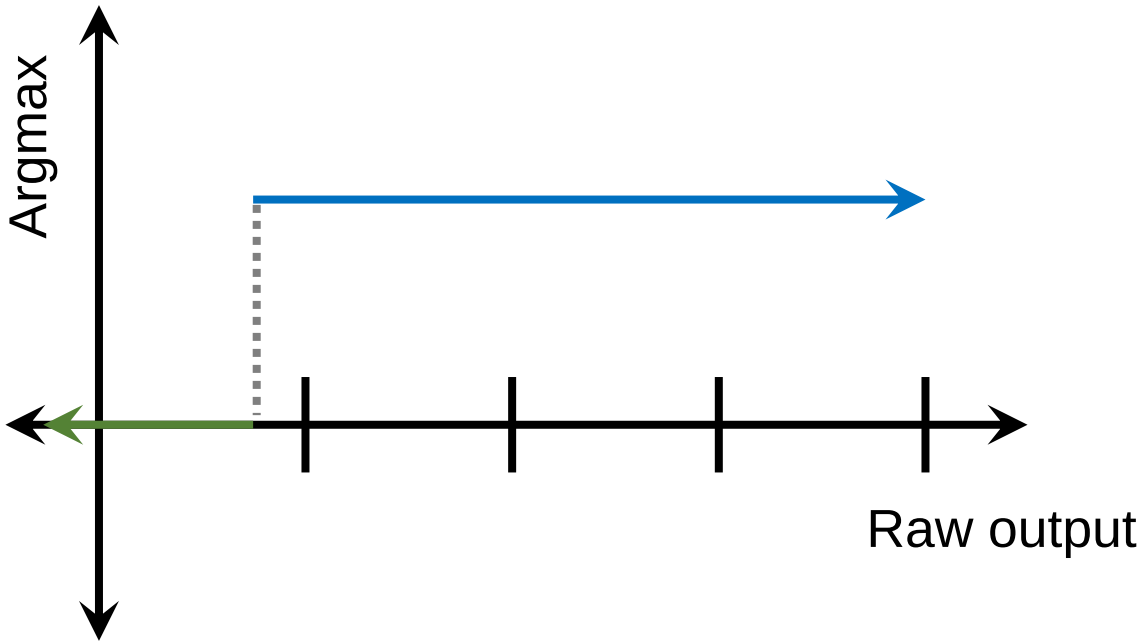
- **Background**
  - When you estimate a deep learning model for categorical outcomes, the raw output could be difficult to interpret.
  - One possible way is to assign 1 to the category having the largest raw output, and 0 otherwise (**Argmax**).
  - Then, you can't use it to optimize the weights and biases in the neural network.



Argmax, Softmax

- **Background**

$$\frac{\partial\ \textbf{Loss Function}}{\partial\ \textbf{Some Parameter}} = \frac{\partial\ \textbf{Loss Function}}{\partial\ \textbf{Argmax}} \cdot \underbrace{\frac{\partial\ \textbf{Argmax}}{\partial\ \textbf{Raw Output}}} \cdot (\dots)$$



Argmax is insensitive to raw outputs, so this derivative is likely 0.

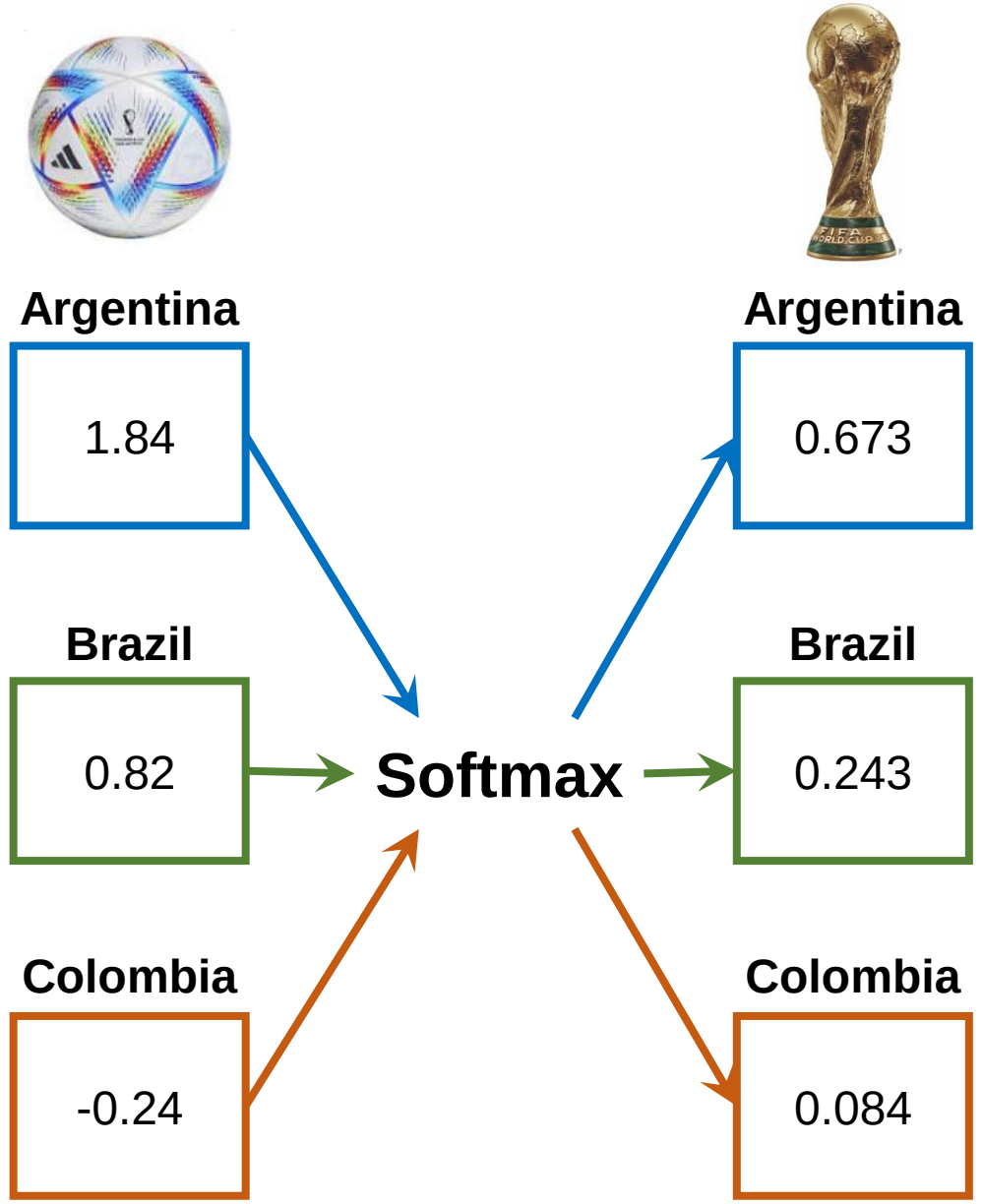
Plugging 0 into the chain rule leads to 0 for gradient descent, so you can't step towards the optimal parameter values.

Argmax, Softmax

- **Softmax Function**
  - The softmax formula is as follows:
    - $s_i = \frac{e^{z_i}}{\sum_{l=1}^{n} e^{z_l}}$
    - Sigmoid : Softmax = Logit : Multinomial Logit
  - The softmax function was developed as a **smoothed** and **differentiable** alternative to the argmax function.
  - It is common to train a machine learning model using the softmax but switch out the softmax layer for an argmax layer when the model is used for inference.
  - Typically, raw outcomes themselves cannot be interpreted as probabilities. If we add a softmax layer to the network, it is possible to translate the numbers into a probability distribution.

Argmax, Softmax

- **Softmax Function**
  - $\mathrm{P}(\text{Argentina} = 1) = \frac{\exp(1.84)}{\exp(1.84)+\exp(0.82)+\exp(-0.24)} = 0.673$
  - $\mathrm{P}(\text{Brazil} = 1) = \frac{\exp(0.82)}{\exp(1.84)+\exp(0.82)+\exp(-0.24)} = 0.243$
  - $\mathrm{P}(\text{Colombia} = 1) = \frac{\exp(-0.24)}{\exp(1.84)+\exp(0.82)+\exp(-0.24)} = 0.084$

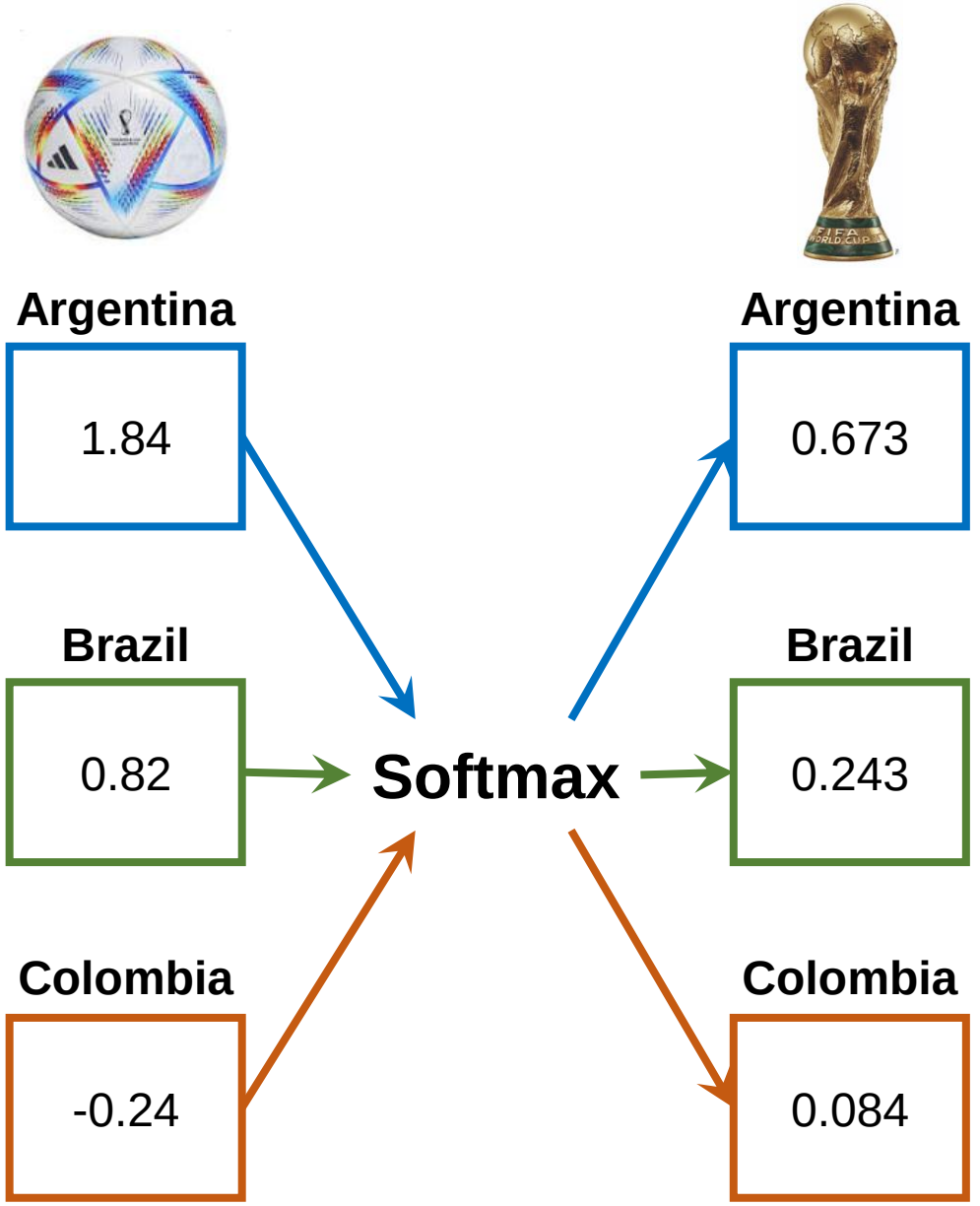


Argmax, Softmax

- **Derivative of Softmax**
  - **Definition:** $s_i = \frac{e^{z_i}}{\sum_{l=1}^{n} e^{z_l}}$
  - **What we want to show:** $\frac{\partial s_i}{\partial z_j} = ?$
  - $\frac{\partial}{\partial z_j} \log(s_i) = \frac{1}{s_i} \cdot \frac{\partial s_i}{\partial z_j}$
    - Derived by the chain rule & $\frac{d}{dx} \log(x) = \frac{1}{x}$
  - $\log(s_i) = \log\left(\frac{e^{z_i}}{\sum_{l=1}^{n} e^{z_l}}\right) = z_i - \log(\sum_{l=1}^{n} e^{z_l})$
  - $\frac{\partial}{\partial z_j} \log(s_i) = \frac{\partial z_i}{\partial z_j} - \frac{\partial}{\partial z_j} \log(\sum_{l=1}^{n} e^{z_l})$
  - Given that $\frac{\partial z_i}{\partial z_j} = 1\{i = j\}$
  - $\frac{\partial}{\partial z_j} \log(s_i) = 1\{i = j\} - \frac{1}{\sum_{l=1}^{n} e^{z_l}} \cdot \frac{\partial}{\partial z_j} \sum_{l=1}^{n} e^{z_l}$
    - Derived by the chain rule & $\frac{d}{dx} \log(x) = \frac{1}{x}$
  - Given that $\frac{\partial}{\partial z_j} \sum_{l=1}^{n} e^{z_l} = e^{z_j}$
  - $\frac{\partial}{\partial z_j} \log(s_i) = 1\{i = j\} - \frac{e^{z_j}}{\sum_{l=1}^{n} e^{z_l}} = 1\{i = j\} - s_j$
  - $\frac{\partial s_i}{\partial z_j} = s_i \cdot \frac{\partial}{\partial z_j} \log(s_i) = s_i \cdot (1\{i = j\} - s_j)$

Argmax, Softmax

- **Derivative of Softmax**
  - Given that $\frac{\partial s_i}{\partial z_j} = s_i \cdot \left(1\{i = j\} - s_j\right)$
  - $\frac{\partial \, \mathrm{P}_{\mathrm{Argentina}}}{\partial \, \mathrm{Raw}_{\mathrm{Argentina}}} = \mathrm{P}_{\mathrm{Argentina}} \cdot \left(1 - \mathrm{P}_{\mathrm{Argentina}}\right) = 0.220$
  - $\frac{\partial \, \mathrm{P}_{\mathrm{Argentina}}}{\partial \, \mathrm{Raw}_{\mathrm{Brazil}}} = \mathrm{P}_{\mathrm{Argentina}} \cdot \left(-\mathrm{P}_{\mathrm{Brazil}}\right) = -0.163$
  - $\frac{\partial \, \mathrm{P}_{\mathrm{Argentina}}}{\partial \, \mathrm{Raw}_{\mathrm{Colombia}}} = \mathrm{P}_{\mathrm{Argentina}} \cdot \left(-\mathrm{P}_{\mathrm{Colombia}}\right) = -0.057$



Cross Entropy

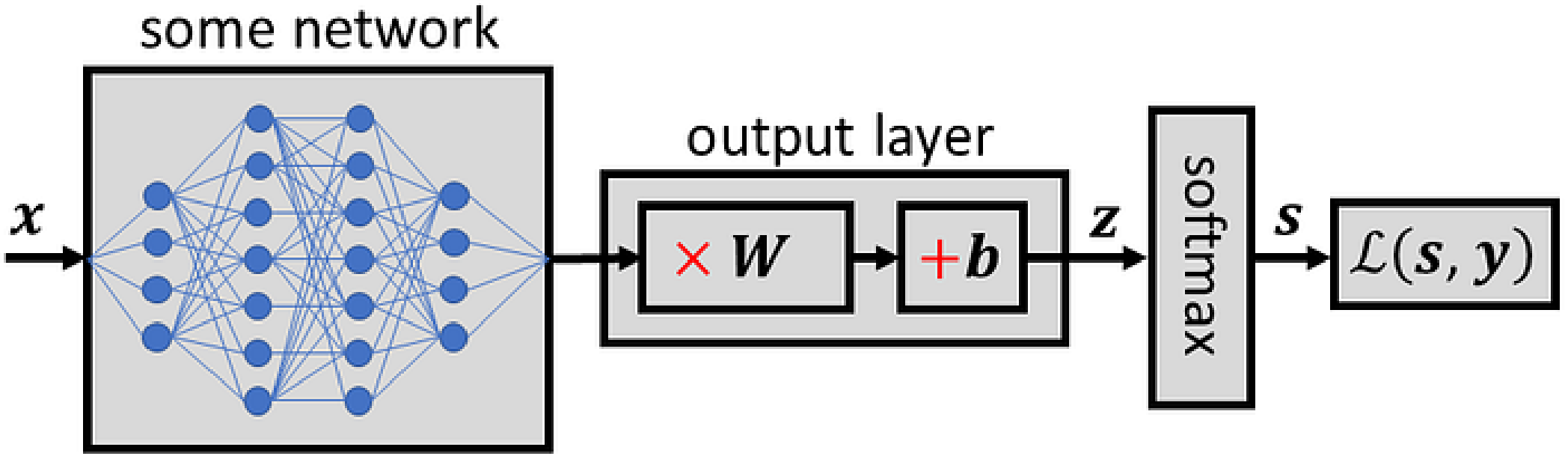
- Classification problems can be subdivided into the following two categories:
  - **multi-class classification**, where each sample belongs to only one class (mutually exclusive)
  - **multi-label classification**, where each sample may belong to multiple classes (or to no class)
- The categorical cross-entropy loss is **exclusively used in multi-class classification tasks**, where each sample belongs exactly to one of the classes.
- Categorical cross-entropy loss is closely related to the **softmax function**, since it's practically only used with networks with a softmax layer at the output.

Cross Entropy

- **Formula**
  - $\mathcal{L}(y, s) = -\sum_{i=1}^{C} y_i \cdot \log(s_i)$
- **Notations**
  - C denotes the number of different classes
  - The subscript i denotes i-th category
- **Interpretation**
  - It takes two discrete probability distributions as input and a single real-valued number representing **the similarity of both probability distributions** as outputs.
  - **The smaller the cross-entropy**, **the more similar** the two probability distributions are.

Cross Entropy

- **Cross Entropy as Loss Function**

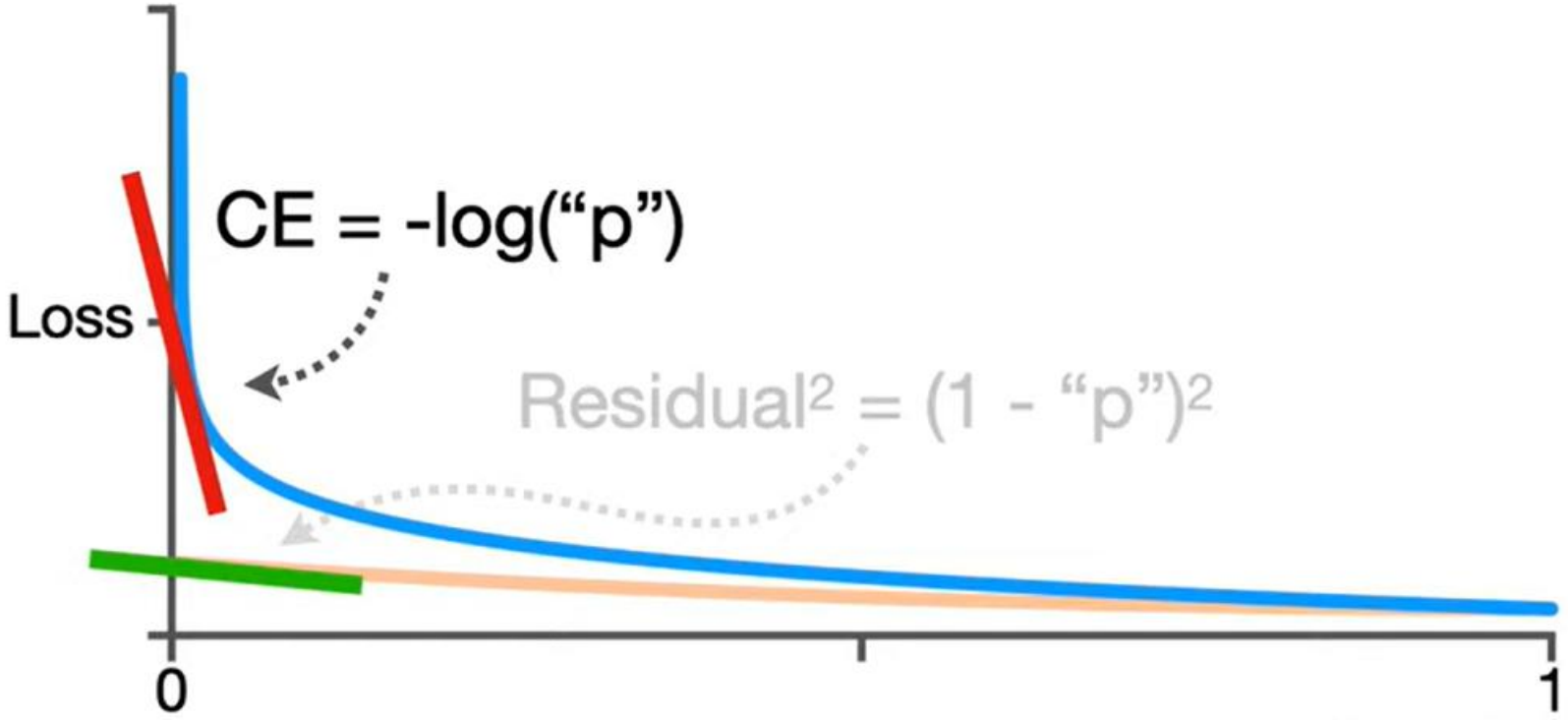


Cross Entropy

- **Gradient for Backpropagation**
  - $\frac{\partial \mathcal{L}}{\partial z_j} = -\frac{\partial}{\partial z_j}\sum_{i=1}^{C} y_i \cdot \log(s_i) = -\sum_{i=1}^{C} y_i \cdot \frac{\partial}{\partial z_j}\log(s_i) = -\sum_{i=1}^{C} \frac{y_i}{s_i} \cdot \frac{\partial s_i}{\partial z_j}$
  - Given that $\frac{\partial s_i}{\partial z_j} = s_i \cdot (1\{i = j\} - s_j)$:
  - $\frac{\partial \mathcal{L}}{\partial z_j} = -\sum_{i=1}^{C} \frac{y_i}{s_i} \cdot s_i \cdot \left(1\{i = j\} - s_j\right) = -\sum_{i=1}^{C} y_i \cdot \left(1\{i = j\} - s_j\right)$
  - By expanding the product in the last term:
  - $\frac{\partial \mathcal{L}}{\partial z_j} = \sum_{i=1}^{C} y_i \cdot s_j - \sum_{i=1}^{C} y_i \cdot 1\{i = j\}$
  - Given that $\sum_{i=1}^{C} y_i = 1$, and $\sum_{i=1}^{C} y_i \cdot 1\{i = j\} = 1$ if $y_j = 1$ and 0 otherwise:
  - $\frac{\partial \mathcal{L}}{\partial z_j} = s_j - y_j$

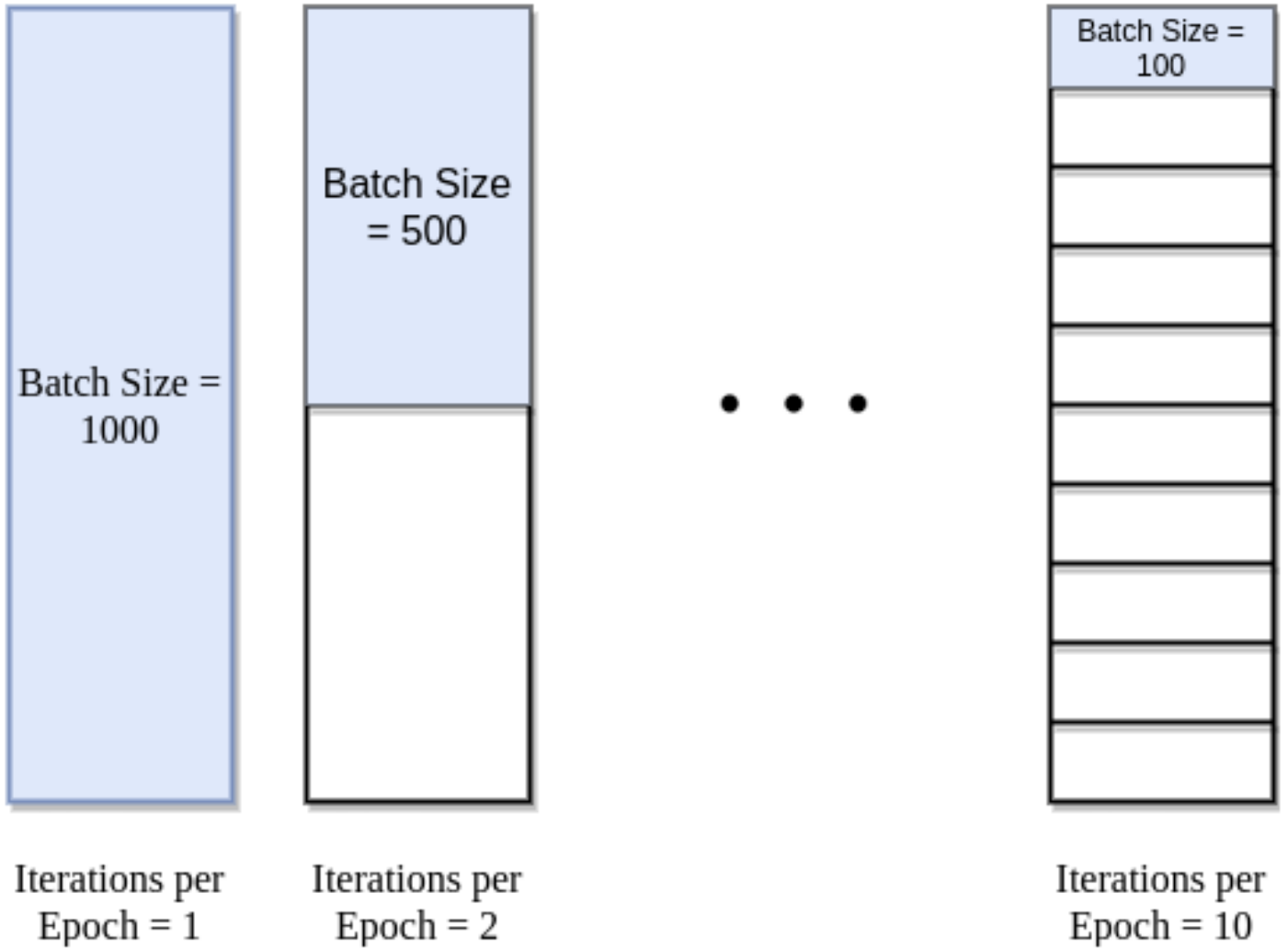
Cross Entropy

- **Gradient for Backpropagation**
  - Cross entropy shows relatively large slopes for a bad prediction compared to squared residuals.



Batch Size

- **Batch** is the unit you divide dataset into. A training dataset can be broken down into multiple batches.
- **Batch size** is the total number of training examples present in a single batch.
- In other words, batch size determines the number of samples taken to work through a particular machine learning model before updating its internal model parameters.

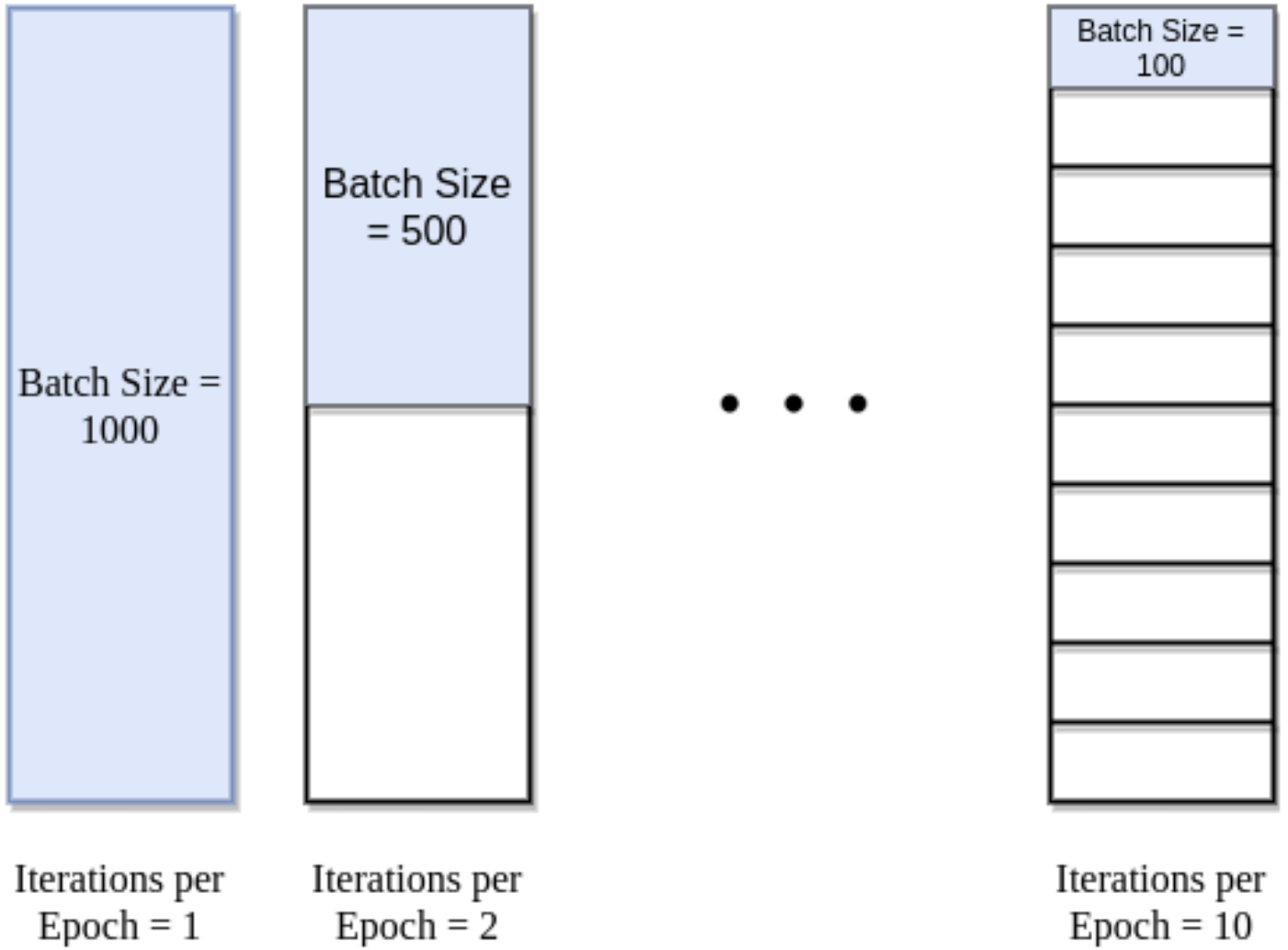


Batch Size

- **Batch size and naming of learning algorithms**
  - **Batch Gradient Descent:** Batch Size = Size of Training Set
  - **Stochastic Gradient Descent:** Batch Size = 1
  - **Mini-Batch Gradient Descent:** 1 < Batch Size < Size of Training Set
  - In the case of mini-batch gradient descent, popular batch sizes include 32, 64, and 128 samples. You may see these values used in models in the literature and in tutorials.
- **What if the dataset does not divide evenly by the batch size?**
  - This can and does happen often when training a model. It simply means that the final batch has fewer samples than the other batches.
  - Alternately, you can remove some samples from the dataset or change the batch size such that the number of samples in the dataset does divide evenly by the batch size.

Epochs

- One **epoch** means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches.
- Another way to define an epoch is the number of passes a training dataset takes around an algorithm. One pass is counted when the data set has done both forward and backward passes.
- **Iterations** is the number of batches needed to complete one epoch.



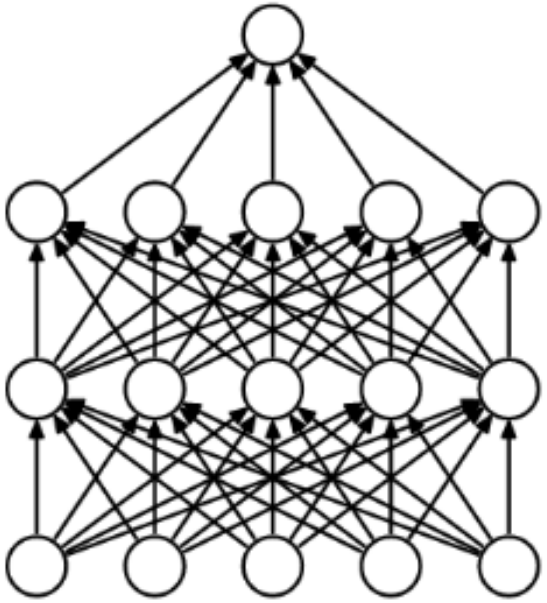
Epochs

- **Difference between a batch and an epoch**

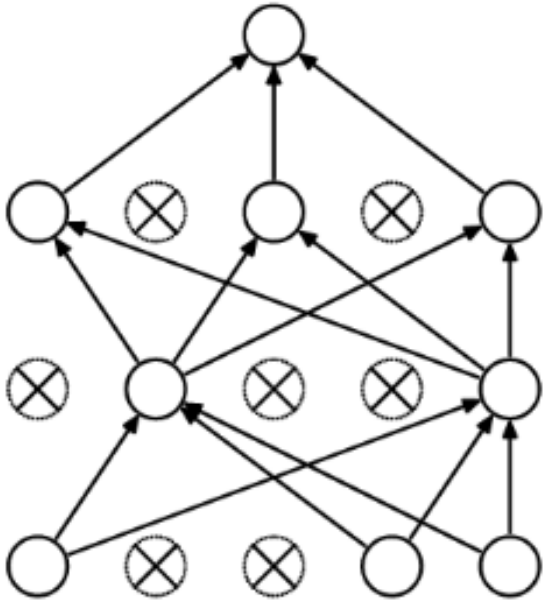
Batch	Epoch
The batch is the dataset that has been divided into smaller parts to be fed into the algorithm.	Epoch is the complete passing through of all the datasets exactly at once.
The batch size is always equal to or more than one and equal to or less than the number of samples in the training set.	The number of epochs can be anything between one and infinity.
It is an integer that is a hyperparameter for the learning algorithm.	It is also an integer value that is a hyperparameter for the learning algorithm.

Dropout

- **What is a dropout?**
  - The term “dropout” refers to dropping out the nodes (input and hidden layer) in a neural network.
  - All the forward and backwards connections with a dropped node are temporarily removed, thus creating a new network architecture out of the parent network.



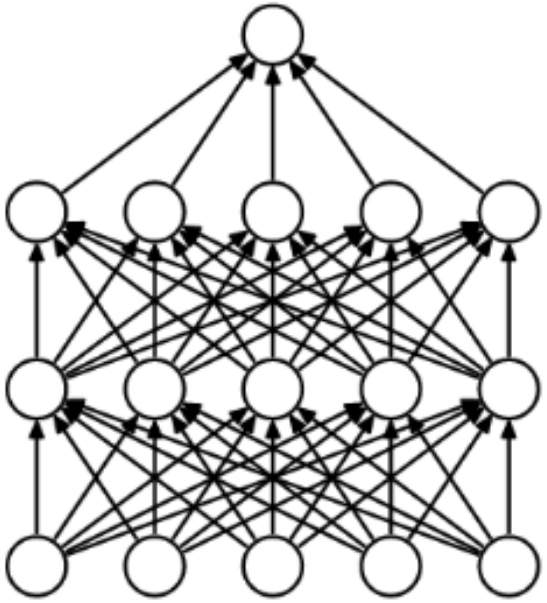
(a) Standard Neural Net



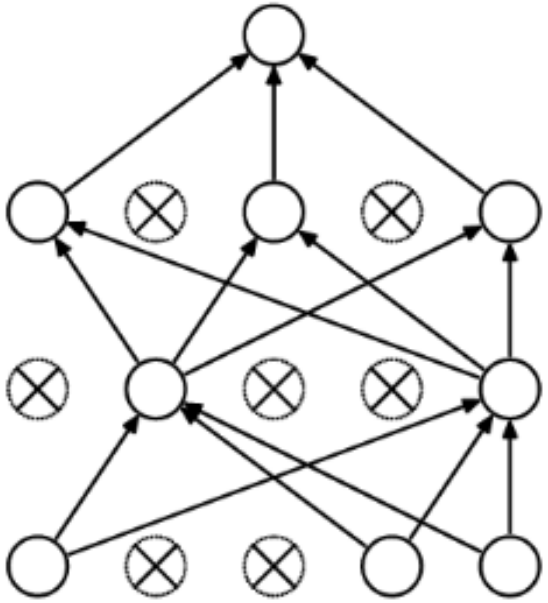
(b) After applying dropout.

Dropout

- **What is a dropout?**
  - The nodes are dropped by a dropout probability of *p*.
  - For instance, if the hidden layers have 1000 neurons (nodes) and a dropout is applied with drop probability = 0.5, then 500 neurons would be randomly dropped in every iteration (batch).



(a) Standard Neural Net



(b) After applying dropout.

Dropout

- **A way to solve the overfitting problem**
  - In this pursuit of trying too hard to learn different features from the dataset, they sometimes learn the statistical noise in the dataset.
  - This improves the model performance on the training dataset but fails massively on new data points.

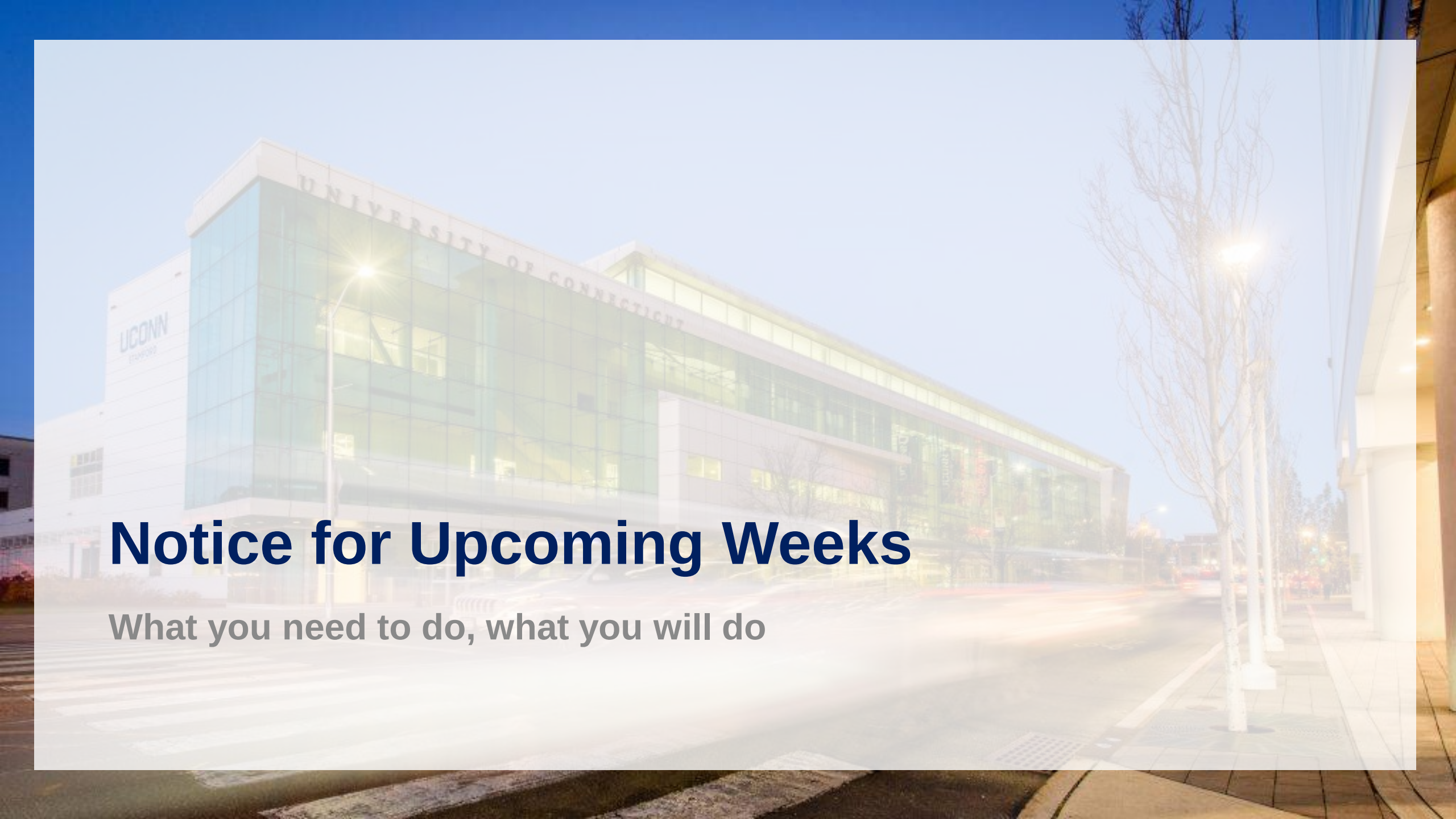
- **How does it solve the problem?**
  - In the overfitting problem, the model learns the statistical noise. In other words, a neuron may change in a way to fix up the mistakes of the other nodes (**co-adaptation**).
  - Dropout prevents neurons to fix up the mistake of other neurons or co-adaptation.
  - By randomly dropping a few neurons (nodes), it forces the layers to take different responsibility for the input by taking a probabilistic approach.

Summary of Hyperparameters

- Number of nodes/neurons
- Number of hidden layers
- Activation functions
- Learning rates
- Epochs
- Batch sizes
- Dropout

References

- ArgMax and SoftMax by **Statquest**
- Cross Entropy by **Statquest**
- What is the Softmax Function? By **DeepAI**
- What is Epoch in Machine Learning? By **Simplilearn**
- Difference Between a Batch and an Epoch in a Neural Network by **Machine Learning Mastery**
- Epoch vs Batch Size vs Iterations by **Towards Data Science**



Notice for Upcoming Weeks

What you need to do, what you will do

What You Will Do

- **Mar 6 (Thu): Mid-term Exam**
- **Mar 13 (Thu)**
  - Feedback for Hands-on Assignment #1
  - Deep Learning: Advanced (Hands-on)
  - Hands-on Assignment #2 is out.
- **Mar 20 (Thu):** No Class (*Spring Recess*)

Jaeung Sim
Assistant Professor
School of Business, University of Connecticut
jaeung.sim@uconn.edu