/images/*

- What is a MaxPooling2D layer? What's it do?
    - A MaxPooling2D layer is a TensorFlow (TF) data structure that implements a neural network (NN) layer. This layer reduces the dimensionality of the inputs, in this case into 2-dimensions. It is known as a Pooling Filter, and max pooling uses the largest value among the inputs (CNNs.pdf, slide 27).
- What's Adam?
    - Adam is an optimizer, an algorithm, for searching the weights for a minimum. It is based on Stochastic Gradient Descent, but gives each weight an individual learning rate.
- What's the softmax function do?
    - The softmax function normalizes a set of real numbers, converting them into a probability distribution, constraining their range between 0 and 1, which allows for the interpretation of values as probabilities where smaller or negative inputs result in lower probabilities and larger inputs correspond to higher probabilities.
- What is CategoricalCrossEntropy? What do we use it for?
    - Entropy is a measure of disorder, much like the more well known physics concept. Measured between 0 and 1, the goal in tuning a model is to bring this disorder (“loss”, or “error”) as low as possible.
- In the CNN example, what does the Flatten layer do?
    - Flattening is another dimensionality reduction technique. In most cases, outputs are reduced to a 1-dimensional vector.
- In the CNN example, what does the Dense layer do?
    - The TF Dense layer recreates the desired graph edges between nodes, and applies the propagation through some activation function, such as ReLU. Each node in each Dense layer is connected to each of the outputs from the previous layer, or inputs if it is the first layer in a NN.
- In the CNN example, why does the height and width get smaller for each convolutional layer?
    - This dimensional shrinking occurs because of the pooling filter applied in the MaxPooling2D layer. Each layer is then passed the dominating pixel or value from the previous layer, and this process is repeated for each layer. This reduces computational complexity.

/load_data/csv

- What does it mean to normalize the data? Where else have we seen normalization?
    - Normalizing a dataset, or features of a dataset, means rescaling the range of all features into a single range, so that each feature could potentially have equal impact on the trained model.
- Why is it a problem that the Titanic data has different types and ranges? Why did we not have to worry about this with the decision tree?
    - Decision trees compare based on intra-feature splits, but don’t compare inter-feature splits. This means that the values in one feature were irrelevant to

the detection of split in another feature for decision trees. However, NNs train based on desired outcomes, not given features. So the scale should be the same inorder to minimize bias of individual extreme inputs. This is what normalization achieves by providing a single scalar range for all values across all features.

- What is a one-hot vector?
    - Also known as a “one-hot encoding”, this is a method of turning non-numeric values into numbers by splitting each of the feature’s values into new columns, and the new values are binary 0 or 1 representing presence at that cell.
- The example that shows how to manually slice the feature dictionary uses yield instead of return. Why is this? What’s the difference between them, and why would you want to use yield?
    - Return exits and terminates a function’s execution immediately.
    - Yield pauses the function’s execution after producing a single slice of some result, maintaining the function’s state.

/text_classification_rnn

- As we know, encoding is a particularly important part of working with neural networks. Explain how text is encoded for an RNN.
    - Raw text from TFDS requires preprocessing. The TextVectorization layer is created and fed with the dataset's text using its .adapt() method to establish vocabulary. The initial 20 tokens are revealed, sorted by frequency after padding and introducing unknown tokens. After establishing the vocabulary, the layer transforms text into indices. Padding is done with zeros to match the longest sequence in the batch. This process is not completely reversible.
- What does the Bidirectional layer do? What are the advantages and disadvantages of this approach? How does it compare to the way we processed sequence data with an HMM?
    - As the name suggests, a Bidirectional layer performs forward and backward propagation. This aids in time-related classification, but is less efficient. HMMs on the other hand, are more efficient, but lack the granularity of time-informed classification.
- What is masking? Why do we need to use it in this example?
    - Masking informs our model of absent timestamps, which helps the model to ignore padding that is part of textual encoding.