**15. Intro to Deep Learning**

**15.1 A Single Neuron**

Using Keras and Tensorflow you'll learn how to:

* create a **fully-connected** neural network architecture
* apply neural nets to two classic ML problems: **regression** and **classification**
* train neural nets with **stochastic gradient descent**, and
* improve performance with **dropout**, **batch normalization**, and other techniques

**What is Deep Learning?**

Some of the most impressive advances in artificial intelligence in recent years have been in the field of *deep learning*. Natural language translation, image recognition, and game playing are all tasks where deep learning models have neared or even exceeded human-level performance.

So what is deep learning? **Deep learning** is an approach to machine learning characterized by deep stacks of computations. This depth of computation is what has enabled deep learning models to disentangle the kinds of complex and hierarchical patterns found in the most challenging real-world datasets.

Through their power and scalability **neural networks** have become the defining model of deep learning. Neural networks are composed of neurons, where each neuron individually performs only a simple computation. The power of a neural network comes instead from the complexity of the connections these neurons can form.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**15.2 Deep Neural Networks**

In this lesson we're going to see how we can build neural networks capable of learning the complex kinds of relationships deep neural nets are famous for.

The key idea here is *modularity*, building up a complex network from simpler functional units. We've seen how a linear unit computes a linear function -- now we'll see how to combine and modify these single units to model more complex relationships.

A screenshot of a computer

Description automatically generated

**Many Kinds of Layers**  
A "layer" in Keras is a very general kind of thing. A layer can be, essentially, any kind of *data transformation*. Many layers, like the [convolutional](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D) and [recurrent](https://www.tensorflow.org/api_docs/python/tf/keras/layers/RNN) layers, transform data through use of neurons and differ primarily in the pattern of connections they form. Others though are used for [feature engineering](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding) or just [simple arithmetic](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Add). There's a whole world of layers to discover -- [check them out](https://www.tensorflow.org/api_docs/python/tf/keras/layers)!

A graph with red dots

Description automatically generated

A graph and diagram on a white background

Description automatically generatedReLU

A diagram of a network

Description automatically generated

**15.3 Stochastic Gradient Descent**

A graph on a white background

Description automatically generated

A white background with black text

Description automatically generated

A white paper with black text

Description automatically generated



A close up of a text

Description automatically generated

A screenshot of a web page

Description automatically generated

**15.4 Overfitting and Underfitting**

A graph on a white background

Description automatically generated

A screenshot of a graph

Description automatically generated

A white background with black text

Description automatically generated

A diagram of a graph

Description automatically generated with medium confidence

**15.5 Dropout and Batch Normalization**