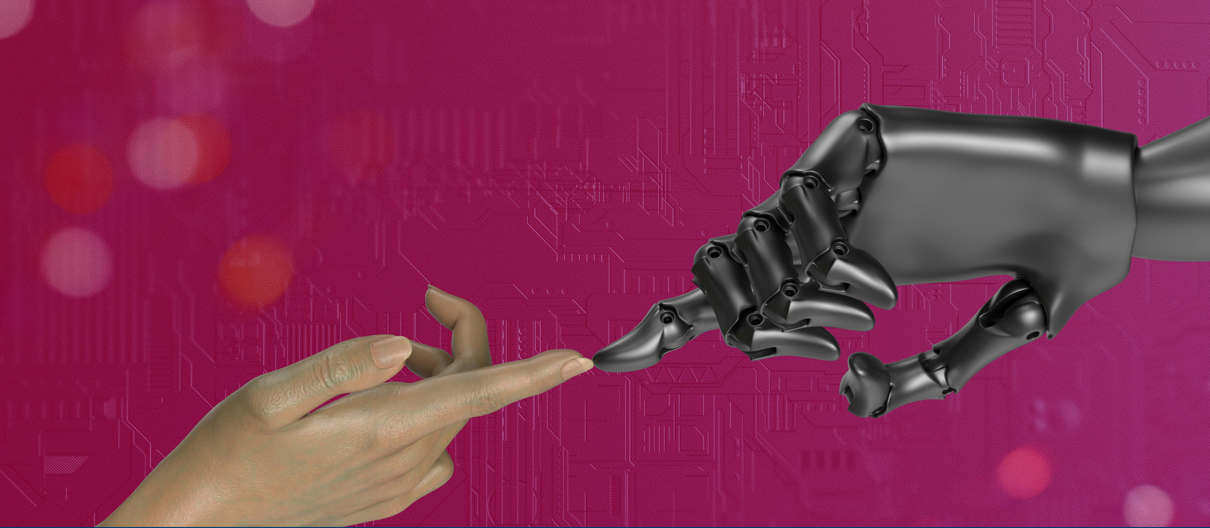


Hands-On Machine Learning

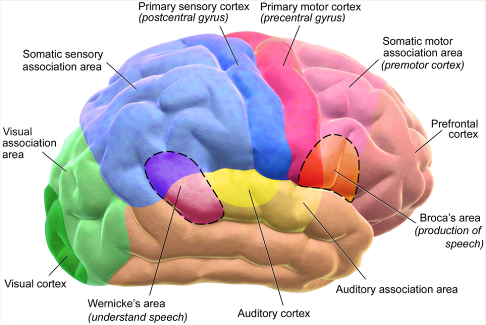
Dr. Derek Bridge | 2026/2027

School of Computer Science and Information Technology
Scoil na Ríomheolaíochta agus na Teicneolaíochta Faisnéise

15. Neural Networks

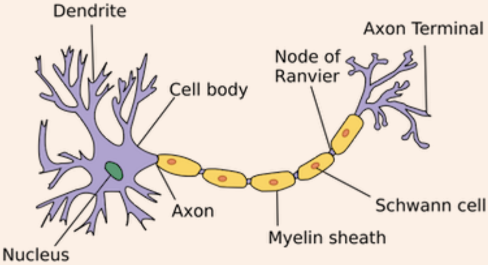
In this lecture, we use three datasets that we have used previously: housing (for regression), CS1109 (for binary classification) and Iris (for multiclass classification).

Brains



Your **brain** is a network of about 10^{11} **neurons**, each connected to about 10^4 others.

Neurons

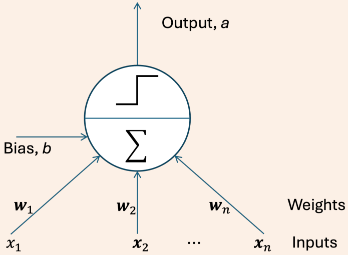


Sufficient electrical activity on a neuron's dendrites causes an electrical pulse to be sent down the axon, where it may activate other neurons.

Neural Networks

- **Neural Networks** are loosely inspired by what we know about our brains.
- However, they are *not* models of our brains — too many simplifications.

Artificial Neurons



- An artificial neuron has n real-valued **inputs**, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$
- The connections have real-valued **weights**, $w_1, \ldots, w_n$
- The neuron also has a number b called the **bias**.
- The neuron computes the weighted sum of its inputs and adds b:

$$z = b + \boldsymbol{w}_1\boldsymbol{x}_1 + \boldsymbol{w}_2\boldsymbol{x}_2 + \cdots + \boldsymbol{w}_n\boldsymbol{x}_n$$

- The neuron then applies an activation function, g, to the weighted sum to produce output a:

$$a = g(z)$$

Activation Functions

- **linear activation function**:

$$g(z) = z$$

- **step activation function**:

$$g(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

- **sigmoid activation function**:

$$g(z) = \frac{1}{1 + e^{-z}}$$

- **ReLU activation function** (ReLU stands for Rectified Linear Unit):

$$g(z) = \max(0, z)$$

- **tanh activation function** (tanh is the hyperbolic tangent):

$$g(z) = \tanh(z)$$

Questions

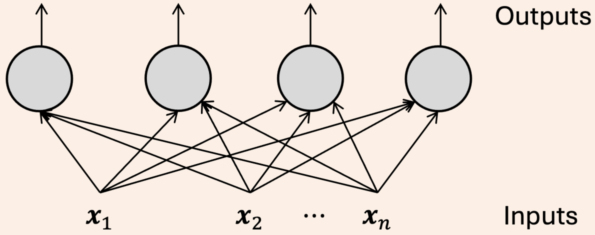
- If we create a model that comprises a single neuron with the linear activation function, then what kind of model do we have?
- If we create a model that comprises a single neuron with the sigmoid activation function, then what kind of model do we have?

Non-Linearity

- Apart from the linear activation function, these activation functions are non-linear, which is important to the power of neural networks.

Dense Layers

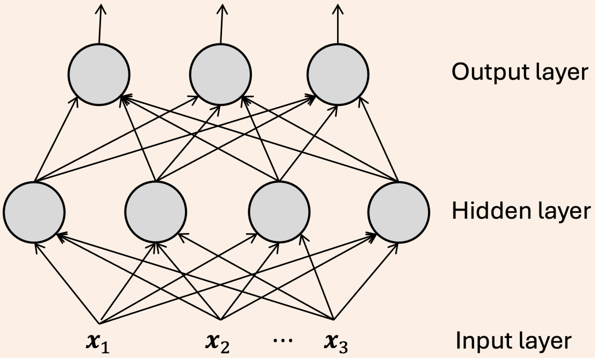
- We don't usually have just one neuron. We have a **layer**, containing several neurons.
- Below is a **dense layer** (also called a **fully-connected layer**) — every input is connected to every neuron in the layer.



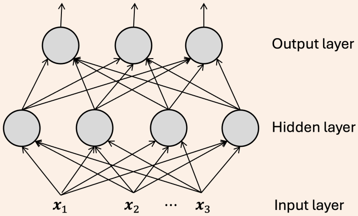
- So now we have more than one output, one per neuron. But each is calculated in the same way as before: compute a weighted sum of the inputs; apply the activation function to the weighted sum.

Dense Layers

- We don't usually have just one layer.
- Let's assume they are also **dense layers**, e.g.:



- The outputs of one layer are the inputs of the next layer.



These neural networks contain:

- an **input layer** (although this is not a layer of neurons);
- one or more **hidden layers**;
- an **output layer**.

Depth

- The depth of a multi-layer neural network is simply the number of layers of *neurons*.
- Question: What is the depth of the network in the diagram?

Deep Networks

- The word ‘deep’ in **deep network** does not mean profound.
- In deep networks, we have ‘lots’ of layers — tens or even hundreds.
- **Deep learning** means training a deep network.

Learning

- Brains learn by strengthening or weakening the connections between biological neurons.
- Analogously, neural networks learn by modifying the values of the parameters (weights and biases).

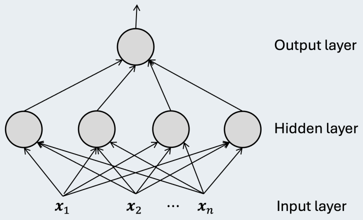
Parametric Learning

- A neural network is just a function — a parameterized function, where the parameters are the weights and biases of the neurons.
- We must choose the network architecture — this is not learned.

Supervised Learning

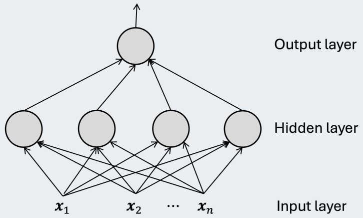
- We must obtain a labelled dataset.
- We must choose the values of numerous hyperparameters (perhaps using, e.g., grid search).
- We must choose a loss function (mean-squared error, binary cross-entropy or categorical cross-entropy, for example).
- Then we run a variant of Gradient Descent, known as **backpropagation**, to find the values of the network's parameters.

A Neural Network for Regression



- Input layer: one input per feature.
- Hidden layers: one or more hidden layers.
- Activation function for neurons in hidden layers can be the sigmoid function or ReLU.
- Output layer: just one output neuron.
- Activation function for the output neuron should be the linear function.

A Neural Network for Binary Classification

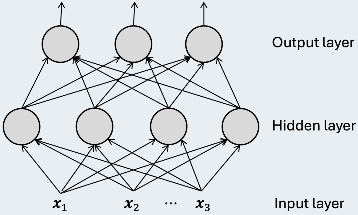


- Input layer: one input per feature.
- Hidden layers: one or more hidden layers.
- Activation function for neurons in hidden layers can be the sigmoid function or ReLU.
- Output layer: just one output neuron.
- Activation function for the output neuron should be the sigmoid function.

Question

Suppose the network outputs 0.7, what will this mean? (Hint: same as Logistic Regression.)

A Neural Network for Multi-Class Classification



(Here we're assuming three classes.)

- Input layer: one input per feature.
- Hidden layers: one or more hidden layers.
- Activation function for neurons in hidden layers can be the sigmoid function or ReLU.
- Output layer: one output neuron per class.
- Activation function for the output neurons should be the softmax function.

Questions

- Why softmax?
- Suppose the network outputs $\langle 0.2, 0.5, 0.3 \rangle$, what will this mean? (Hint: same as Multinomial Logistic Regression.)

Model Complexity

- More layers —and more neurons in each layer— increase model complexity.
- Obvious — because it increases the number of parameters (weights and biases).

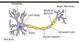
Representation learning

- A neural network takes a vector of data into its input layer.
- The first layer in some sense transforms the input vector into a new vector — a different representation of the input vector.
- The second layer transforms the output of the previous layer into another new vector — another representation; and so on.
- So, learning in neural networks is about finding successive layers of representations.

Automated Feature Engineering

- In Feature Engineering, we used our ingenuity to derive new features from existing features.
- In a neural network, a neuron combines the outputs of previous neurons — a bit like computing a new feature from existing features.
- So, learning in neural networks automates feature engineering — however, these new features are not easily interpretable.

Image credits

	Taken from Wikipedia
	Taken from Wikipedia