Hi, I’m here to talk about the applications of neural networks. There are many places where neural networks can be used, and some examples are displayed here, which I will talk about in a bit, but first of all I’d like to talk about some of the types of application you can get, since most applications will fall under one of these categories.

So first of all we have classification. This is when a neural network can categorise inputs to a specific class, and this is done by using common patterns in the input data - so this data needs to be accurate, otherwise you’ll end up training the network for the wrong thing and this won’t give your desired result. The network can show which class it has chosen for an input depending on the value of an output node; each node will correspond to a different class, so if one is higher by a significant amount then the network thinks it belongs to that class.

Next up is clustering. This is quite similar to classification in that it groups stuff together, but it differs because it uses unsupervised learning, and works out what the clusters should be from the data rather than you giving it the classes, which would be classification. Also, clustering is commonly used in situations where you can separate things using distances.

Pattern association; there are two types of this, auto-association and hetero-association. Auto-association is when you remove noise or corruption from an input, say an image, so this is basically trying to recreate a better version of the input you give it. Hetero-association is associating an input with a certain output, say a face to a name.

Vector quantising - you can convert an input into vectors, and then associate a bunch of these to a similar vector in way that's similar to clustering; this allows you to compress a lot of input data although it isn't lossless since you've removed information.

So, the first application we have is optical character recognition (i.e. turning an image with words into actual words) – algorithms for this used to have to use a lot of processing power, since they used things like smoothing and curve analysis. This took a lot of processing power for the time, and could cause delays in recognition time, which wasn't good for commercial environments where speed is crucial.

So, neural networks were seen as an alternative since they could do without so much processing. They could actually be used with the back propagation algorithm since you can give it ideal examples, and while this does have the downside of trying to get the initial learning environment correct, that's one time only and once it's done it's a fairly effective way of recognising characters efficiently.

Image compression – this can be done with neural networks because you can decompose an image into vectors, using brightness or colour. Once this is done, you can use vector quantisation, and that has some different algorithms such as competitive learning or using a self-organising feature map. There are actually other ways of compressing an image aside from vector quantisation that might be lossless, which is good for the image but might not be able to compress as much. (HEBBIAN LEARNING/PREDICTIVE CODING/ADAPTIVE BACK PROPAGATION)

And finally we have predicting the stock market. The first question to ask for this is why we should use neural networks at all, since there's already established techniques for this. Well as far as I know, these rely on a lot of analysis of existing data to find patterns, and neural networks are pretty adept at finding complicated, non-linear patterns, using multiple hidden layers.

When designing a network for this, you have to look at how many input neurons to use, since stock markets are quite complex and have many factors affecting them. You also have to look at which learning algorithm is most ideal since this is an important area, and back propagation seems pretty ideal since it's fairly easy to implement and is pretty effective at what it does, and you can give it existing market data to train off.

So that's some of the applications we've looked at. I've mentioned the back propagation algorithm a lot, so here's my colleague Christophe to explain exactly how that works in detail.