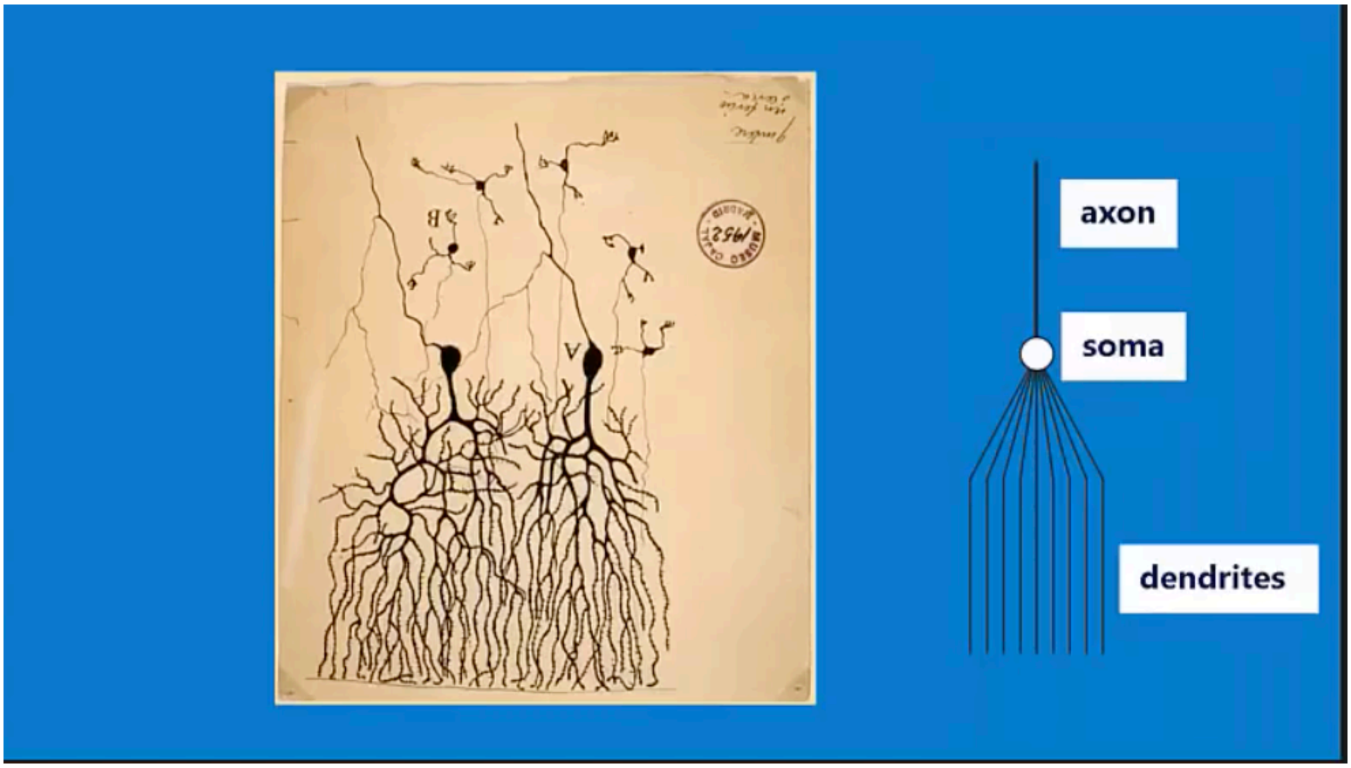
- ***deep learning***
  - -> there is too much hype about deep learning in the media <- it has limitations
  - -> deep learning is good at finding patterns
    - -> highly specialised and engineered
  - **-> neurones**
    - -> each branches off as part of a network
    - -> they are tightly packed / dense (in nature)
    - -> each is connected to multiple neurones
    - -> the first images of these were taken under 19th century microscopes

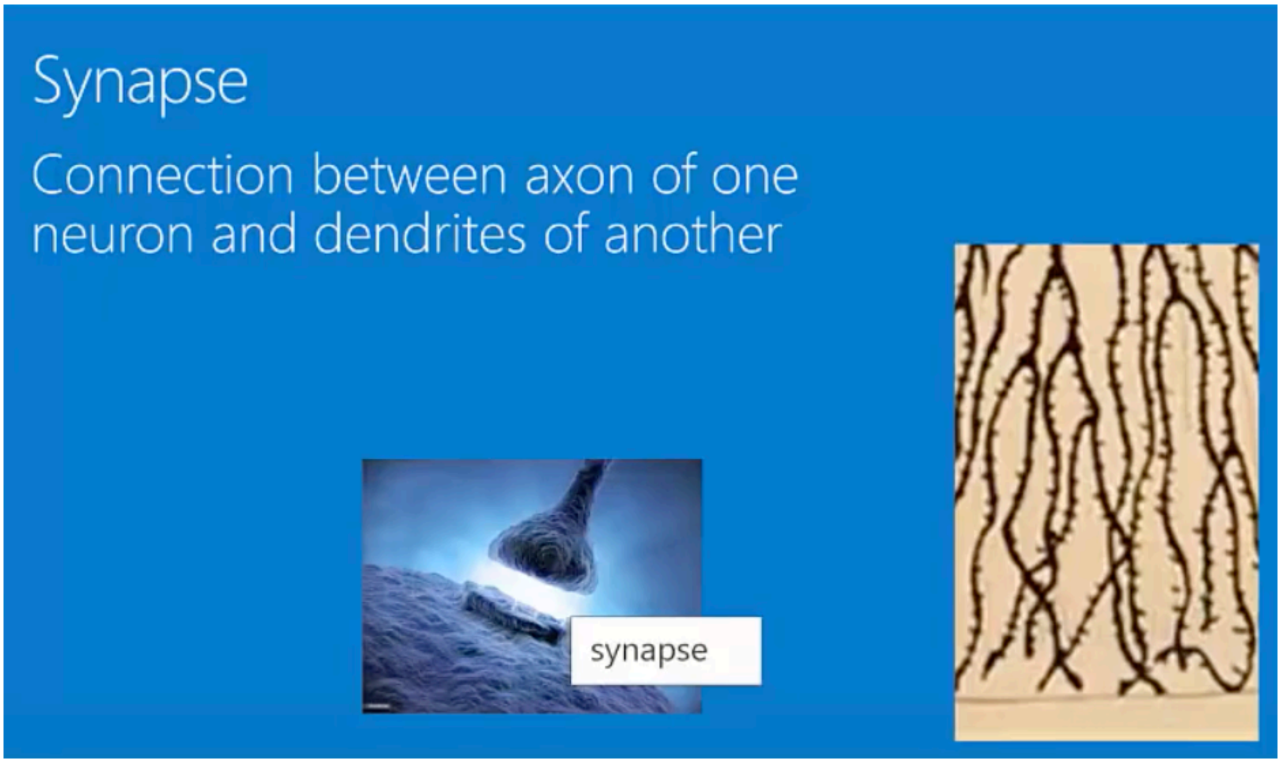


axon

soma

dendrites

    - -> dendrites are the feelers which look for electrical activity
    - -> then the signal is sent along the tail
    - -> the more activity there is the more of the dendrites will be firing
    - ***-> the synapse***
      - -> where the axon of one neurones touches the dendrites of another

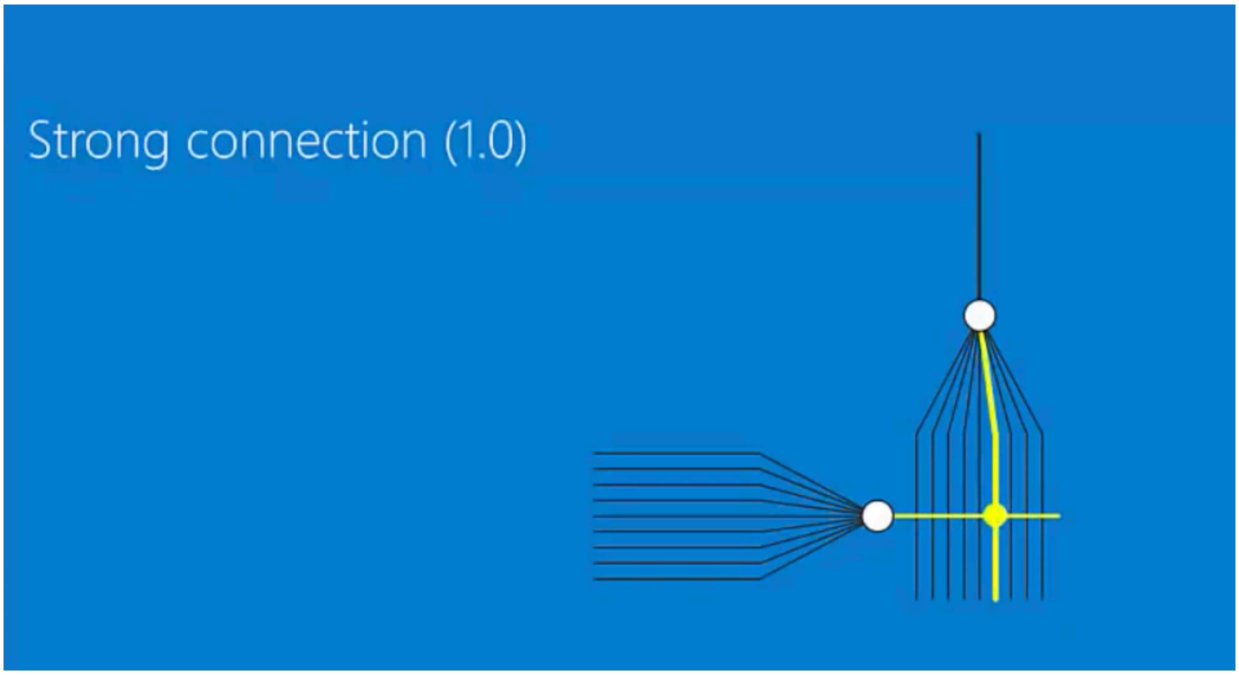


Synapse

Connection between axon of one neuron and dendrites of another

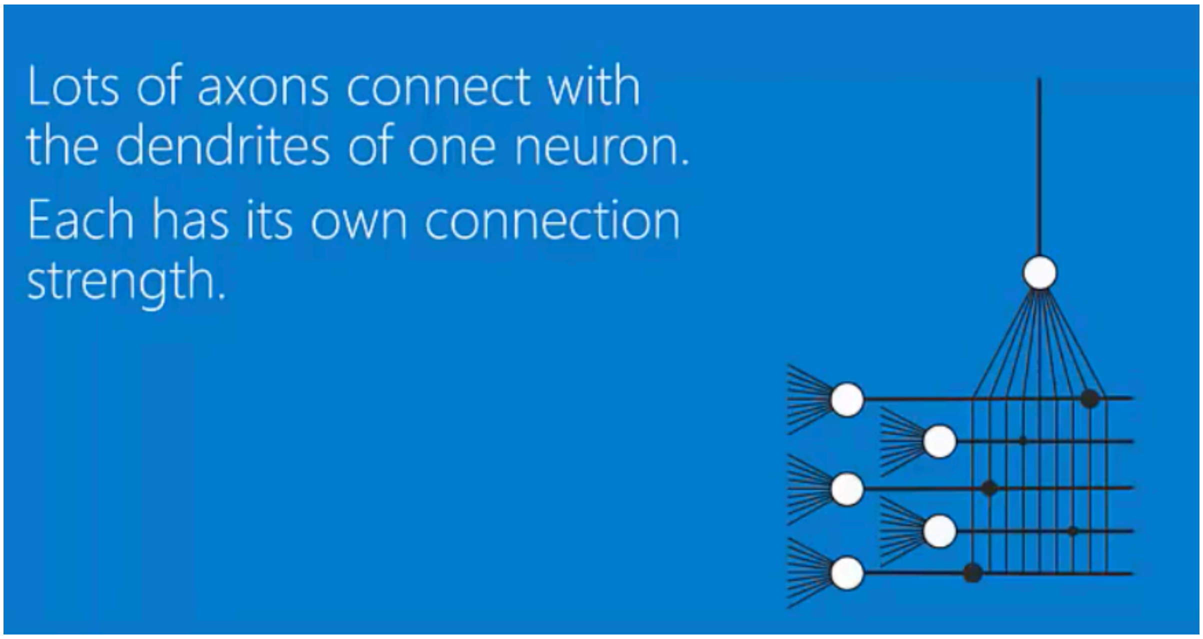
synapse

- ***-> the diameter of the circle is the strength of the connection***



Strong connection (1.0)

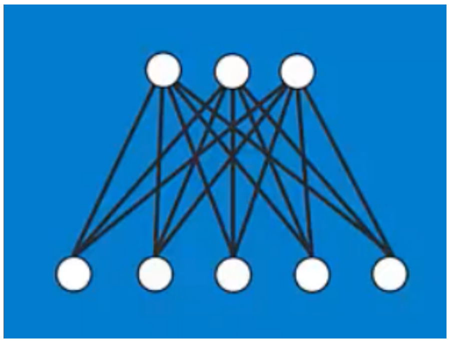
  - -> the connections between them can be a 0 or 1
  - -> no connection is a 0
  - -> you can have many different input neurones to a single dendrite



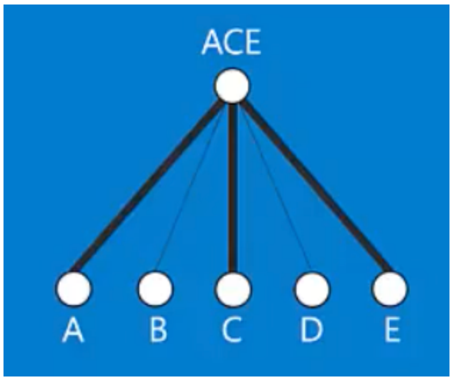
Lots of axons connect with the dendrites of one neuron.

Each has its own connection strength.

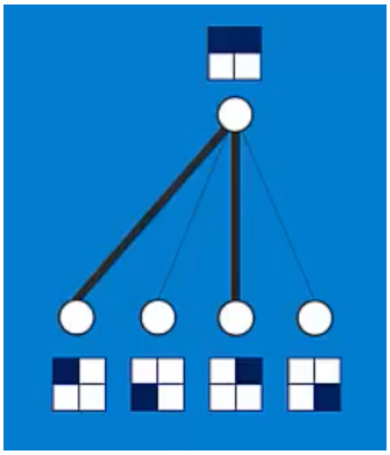
- ***-> each of the lines represents the weight of the connection***



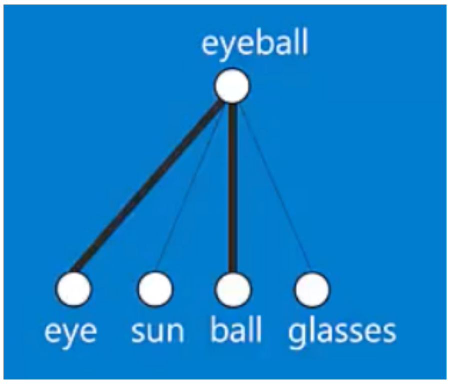
    - -> the connection is distinct and has its own weight



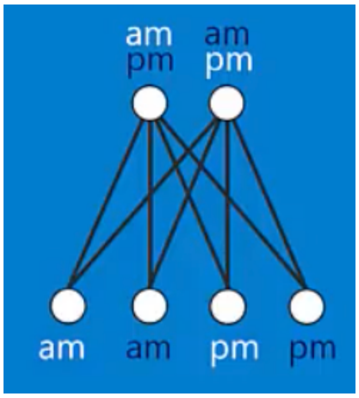
- -> some of the connections can be so strong the others are negligible -> in this example B and D are negligible
- -> you can do this with any kind of input



- -> this model works with pixels as the input



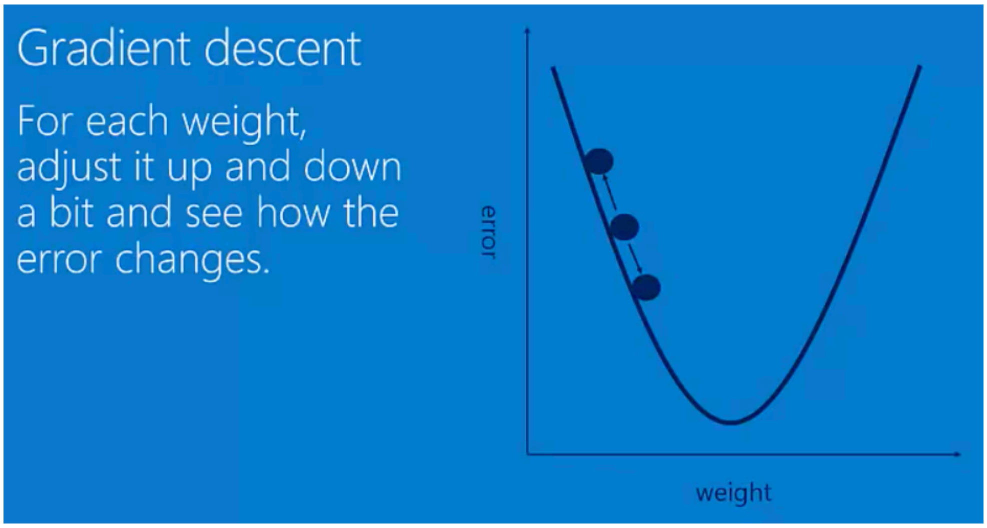
- -> it also works with words as the input



- -> you can have many output neurones
- -> domain knowledge -> e.g working in the mornings some days and evenings on others
- -> representing the working patterns in terms of neurones
- -> neurones can be used to represent different outcomes / possibilities

- ***-> how can we learn the network?***
  - -> choose the number of output neurones and randomly assigning weights
  - -> then we gather data
  - -> so we have the model (e.g the number of neurones) and then we randomly give it weights
  - -> it makes predictions based off of those weights and tunes them while it's being trained
  - -> we are calculating the activity of each of the output neurones -> the average of its inputs

- -> then we calculate the errors based off of the predictions it made
- ***-> then we do gradient descent -> adjusting the weights to improve the model (decreasing the error)***
  - -> it's the relationship between the error and the weight
  - -> we want to reach the local / global minimum for the error of the predictions

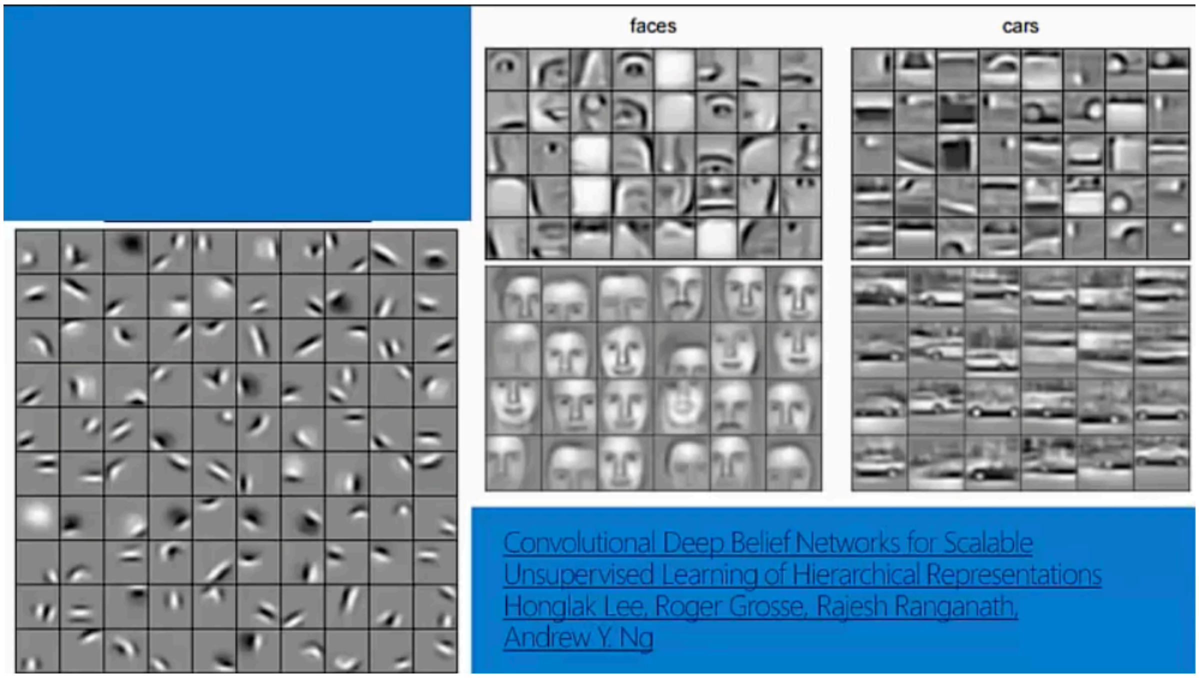


Gradient descent

For each weight, adjust it up and down a bit and see how the error changes.

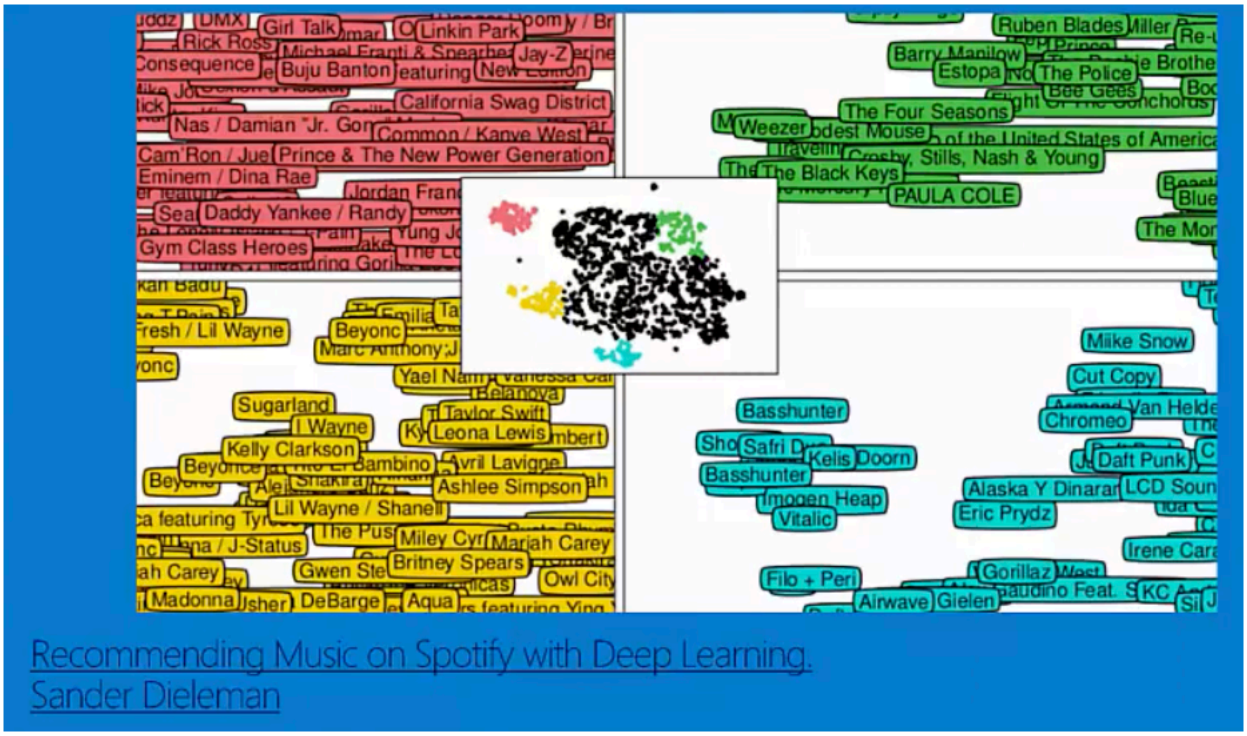
- -> some of the weights affect the predictions of the model more than the others
- ***-> you do this for all of the datapoints in the epochs for the training set and the model improves***
  - -> the more times you train the model the more the weights converge
  - -> back propagation is changing the weights after an iteration of gradient descent

- ***-> training the network with multiple layers***
  - -> the outputs for one layer can be the inputs for the next
  - -> why it's called deep learning (deep)
    - -> for each layer of the network you are getting combinations of the different outputs
      - e.g words in the English language
- ***-> applications***
  - -> images and pixels as the inputs



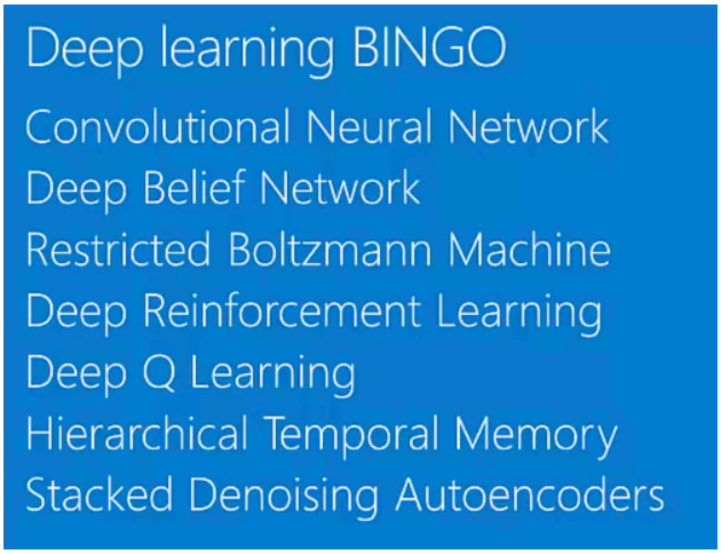
Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng

    - -> the different shapes in the images / lines are the primatives
    - ***-> each of the layers of the networks have different building blocks***
      - -> for example the first layer having all lines
      - -> the second layer starting to form facial features
      - -> it's learned it from the images -> without having to tweak the weights
    - ***-> he has another example of deep vision***
      - -> more complex objects / colours
      - -> information about music artists
        - -> plotting information about how similar the artists are together



Recommending Music on Spotify with Deep Learning.
Sander Dieleman

        - -> it finds patterns and groups things which match those patterns
      - -> re-enforcement actions can also be used to play games
      - ***-> robots can learn about how to cook from YouTube***
        - -> it uses deep neural networks -> to interpret the video, understand its movements - then to execute them itself based off of what it's seen
- ***language***



Deep learning BINGO

Convolutional Neural Network
Deep Belief Network
Restricted Boltzmann Machine
Deep Reinforcement Learning
Deep Q Learning
Hierarchical Temporal Memory
Stacked Denoising Autoencoders

    - -> deep learning learns patterns
    - **-> neurones which are being trained for the first time are initially given random weights in the network**