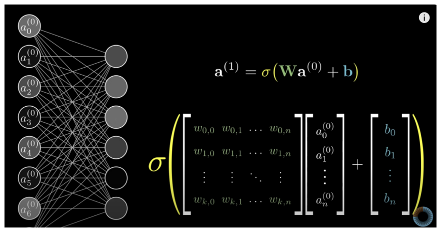
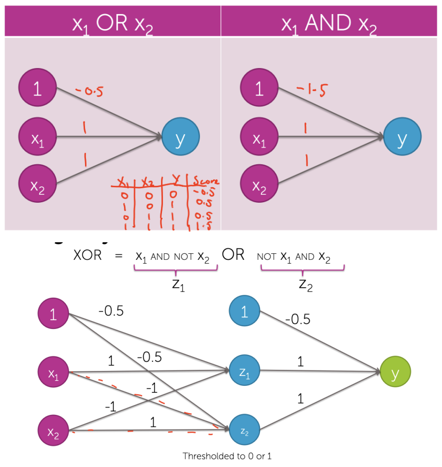
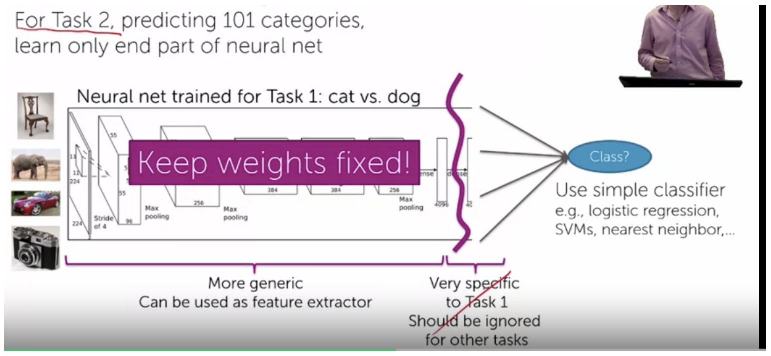
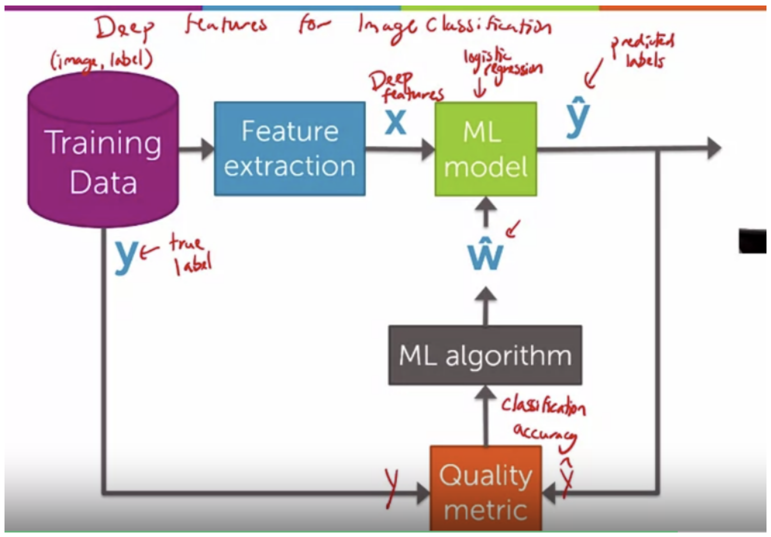
| Deep Learning: Image Search | |
|---|---|
| Neural Networks | Learning very non-linear features and forming them in layers. Activation of one layer triggers a pattern of activation of another layer.<br><br><u>Example</u>: to recognize 9 in the last layer, each layer before it breaks down each edges that can form pieces of 9 until the network pieces it together to be 9.<br><br>So how can we identify which pixels are “lit up” to form an image. What “dials and knobs” can you tweak to see the image?<br>- Each neuron in the first layer is assigned a weight.<br>- Only the weights in the region we care about is nonzero, then taking the weighted sum of the pixels is equal to the sum of the area we care about. Negative weights might exist too to define the edge.<br>- Sigmoid function exists to squish the real number line into a range between 0 and 1 where 1 is the most active.<br><br><u>Example of using Layers:</u><br>- Layer 1: all the pixels in the images<br>- Layer 2: picks up on the edges (where the pixel is active)<br>- Layer 3: picks up on the patterns<br>- Layer 4: output the answer of the image<br><br>The weighted function is really a linear algebra equation where there is a matrix of weights multiplied to the neurons and the bias is added to it.<br><br>$\mathbf{a}^{(1)} = \sigma(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b})$<br>$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right)$<br> $a_0^{(0)}$ $a_1^{(0)}$ $a_2^{(0)}$ $a_3^{(0)}$ $a_4^{(0)}$ $a_5^{(0)}$ $a_6^{(0)}$ |

| | |
|---|---|
| | Linear Classifiers can represent OR and AND. In order to also show XOR, we use a second layer and this is why we have layers in neural networks.<br> $x_1$ OR $x_2$<br>$x_1$ AND $x_2$<br>1<br>$x_1$<br>$x_2$<br>y<br>-0.5<br>-1.5<br>XOR = $x_1$ AND NOT $x_2$ OR NOT $x_1$ AND $x_2$<br>$z_1$<br>$z_2$<br>Thresholded to 0 or 1 |
| Application of Deep Learning | Historically identification is done by hand.<br>- The program identifies unique features such as corners.<br>- It stores these features into a vector.<br>- It pushes this vector into a classifier which identifies what the image is.<br><br>Challenge is handcrafting is complicated process, but neural network does this automatically through layers<br>- Layer 1: detectors identify left diagonal edge, right diagonal edge, and transition in color<br>- Layer 2: detectors identify patterns and corners<br>- Layer 3: detectors react to torso and faces<br>- Layer 4: makes a prediction |
| Deep Learning Examples | Image Parsing<br>- Scene understanding, identifying trees, roads, clouds…<br>Retrieving Similar Images |
| Challenges of Deep Learning | Pro<br>- Enables learning of non-linear items, features than hand tuning<br>- Impressive performance gains<br>- Potential for more impact<br>Cons<br>- Lots of labeled data (training and validation)<br>- Computationally expensive<br>- Hard to tune |
| Deep Features | Prediction even with little data or time |

| | |
|---|---|
| | Transfer learning: use data from one task to help learn on another.<br>- The layers near the beginning of the neural networks are generic enough to be applied to the other task we want. Later layers may be too specific and should be ignored. We just have to add on a classifier at the end for this to work on the new task.<br> For Task 2, predicting 101 categories,<br>learn only end part of neural net<br>Neural net trained for Task 1: cat vs. dog<br>Keep weights fixed!<br>Class?<br>Use simple classifier<br>e.g., logistic regression, SVMs, nearest neighbor,...<br>More generic<br>Can be used as feature extractor<br>Very specific to Task 1<br>Should be ignored for other tasks |
| Deep Learning ML Block Diagram |  Deep features for Image Classification<br>(image, label)<br>Training Data<br>Feature extraction<br>Deep features<br>x<br>logistic regression<br>ML model<br>ŷ<br>predicted labels<br>y ← true label<br>ŵ<br>ML algorithm<br>classification accuracy<br>Quality metric<br>y<br>ŷ |