Coursera
Deep Learning Specialization
Week 3

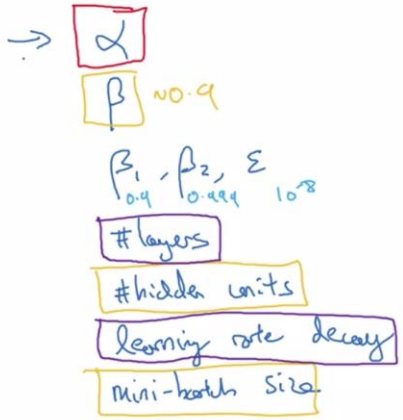
collaged by

Kivanc Babacan

Hyperparameter tuning, Batch Normalization and Programming Frameworks

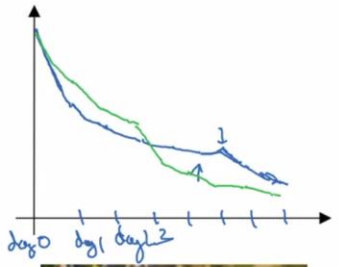
1. Tuning process
2. Using an appropriate scale to pick hyperparameters
3. Hyperparameters tuning in practice: Pandas vs. Caviar
4. Normalizing activations in a network
5. Fitting Batch Norm into a neural network
6. Why does Batch Norm work?
7. Batch Norm at test time
8. Softmax Regression
9. Training a softmax classifier
10. Deep learning frameworks
11. TensorFlow

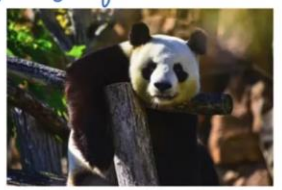
Hyperparameters



Andrew Ng

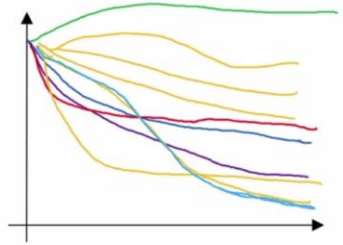
Babysitting one model





Panda

Training many models in parallel

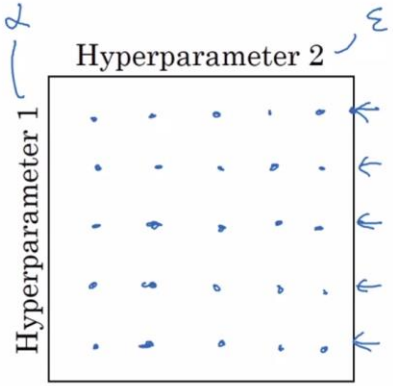


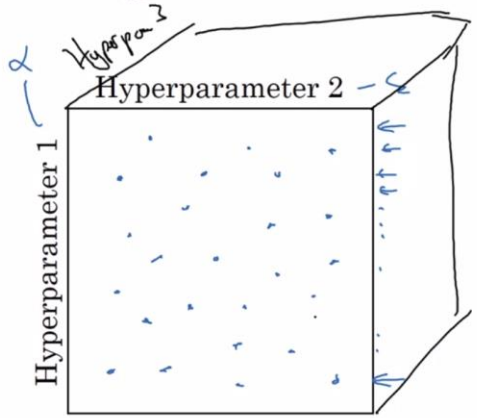
Caviar

Andrew Ng

HYPERPARAMETERS

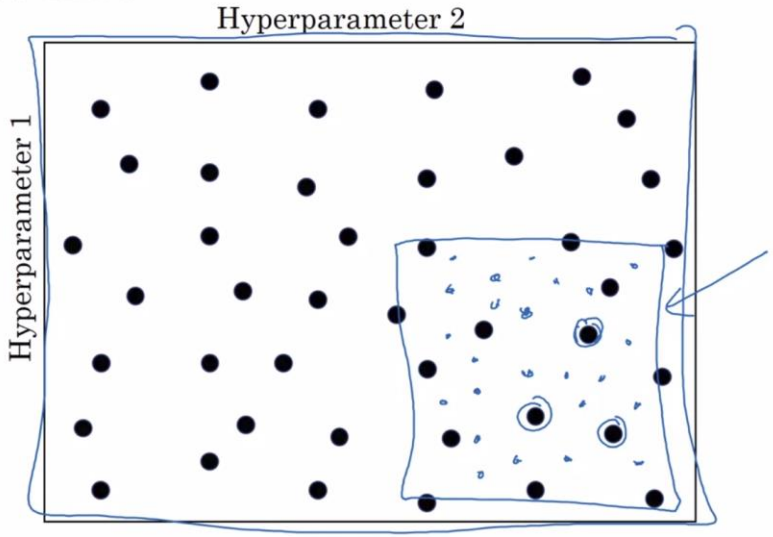
Try random values: Don't use a grid





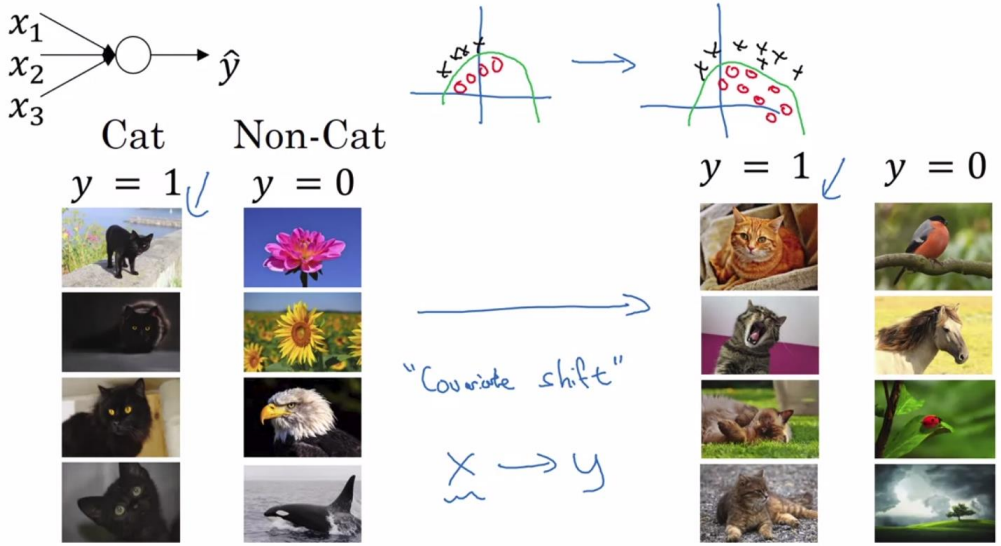
Andrew Ng

Coarse to fine



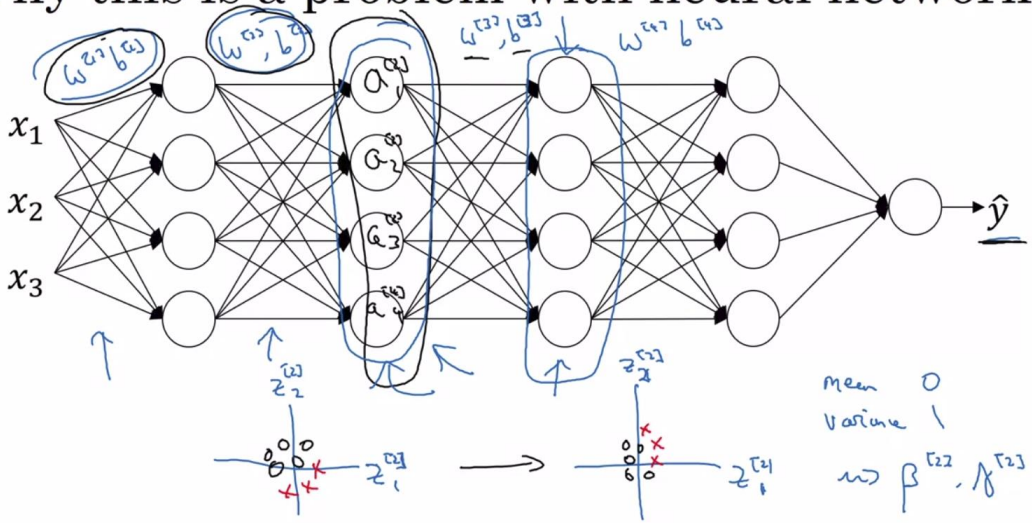
Andrew Ng

Learning on shifting input distribution



Andrew Ng

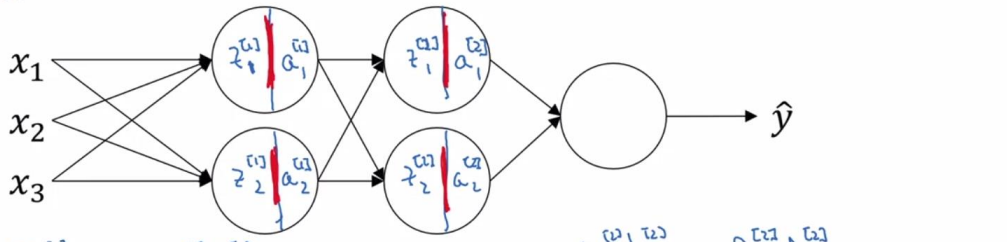
Why this is a problem with neural networks?



Andrew Ng

BATCH NORMALIZATION

Adding Batch Norm to a network



Andrew Ng

Batch Norm at test time

$$\mu = \frac{1}{m}\sum_i z^{(i)}$$

$$\sigma^2 = \frac{1}{m}\sum_i (z^{(i)} - \mu)^2$$

$$z^{(i)}_{\text{norm}} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$$\tilde{z}^{(i)} = \gamma z^{(i)}_{\text{norm}} + \beta$$

μ, σ^2: estimate using exponentially weighted average (across mini-batches).

$X^{\{1\}}, X^{\{2\}}, X^{\{3\}}, \ldots$

$\mu^{\{1\}[l]}$ $\mu^{\{2\}[l]}$ $\mu^{\{3\}[l]}$ $\rightarrow \mu$

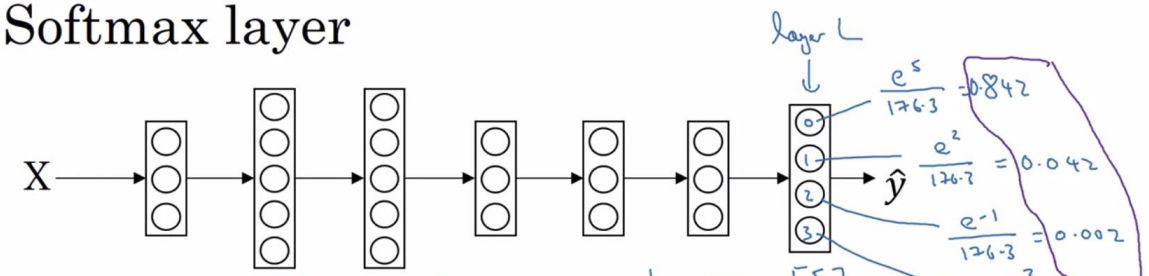
θ_1 θ_2 θ_3 σ^2

$\sigma^{2\{1\}[l]}$ $\sigma^{2\{2\}[l]}$ ....

$z_{norm} = \frac{z - \mu}{\sqrt{\sigma^2 + \varepsilon}}$ $\tilde{z} = \gamma z_{norm} + \beta$

Andrew Ng

Softmax layer



$z^{[L]} = W^{[L]} a^{[L-1]} + b^{[L]}$ (4,1)

Activation function:

$t = e^{(z^{[L]})}$

$a^{[L]} = \frac{e^{z^{[L]}}}{\sum_{j=1}^{4} t_i}$ (4,1), $a_i^{[L]} = \frac{t_i}{\sum_{j=1}^{4} t_i}$

$a^{[L]} = g^{[L]}(z^{[L]})$ (4,1)

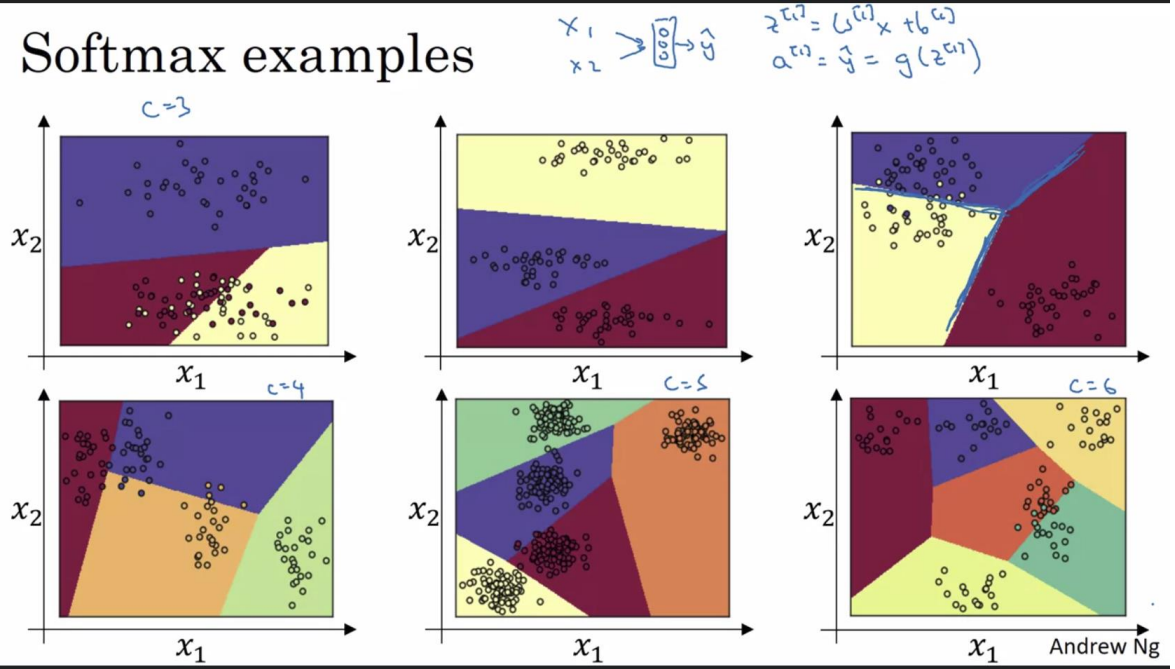
$z^{[L]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$

$t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix} = \begin{bmatrix} 148.4 \\ 7.4 \\ 0.4 \\ 20.1 \end{bmatrix}$ $\sum_{i=1}^{4} t_i = 176.3$

$a^{[L]} = \frac{t}{176.3}$

Andrew Ng

Softmax examples



$x_1, x_2 \rightarrow \hat{y}$

$z^{[L]} = W^{[L]} x + b^{[L]}$

$a^{[L]} = \hat{y} = g(z^{[L]})$

Andrew Ng

SOFTMAX

Loss function

$y^{(i)}$ (4,1) $= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ — cat, $y_2 = 1$, $y_1 = y_3 = y_4 = 0$

$a^{[L]} = \hat{y}^{(i)}$ (4,1) $= \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$ C = 4

$\mathcal{L}(\hat{y}, y) = -\sum_{j=1}^{4} y_j \log \hat{y}_j$ — small

$J(W^{[1]}, b^{[1]}, \ldots) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

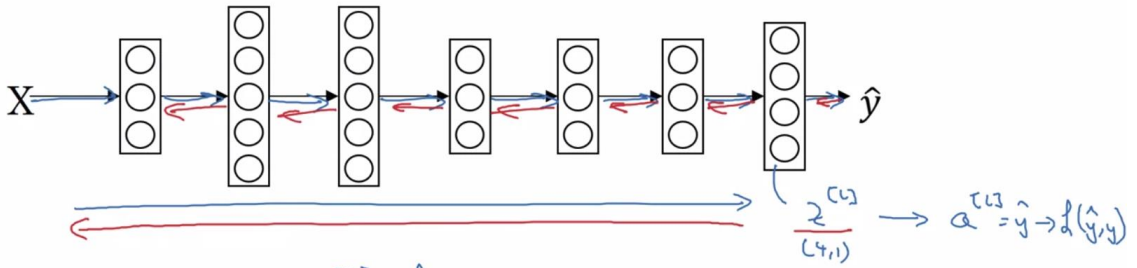
$-y_2 \log \hat{y}_2 = -\log \hat{y}_2$. Make $\hat{y}_2$ big.

$Y = [y^{(1)}\ y^{(2)} \cdots y^{(m)}]$ $= \begin{bmatrix} 0 & 0 & 1 & \cdots \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \end{bmatrix}$ (4, m)

$\hat{Y} = [\hat{y}^{(1)}, \cdots, \hat{y}^{(m)}]$ $= \begin{bmatrix} 0.3 & \cdots \\ 0.2 & \\ 0.1 & \\ 0.4 & \end{bmatrix}$ (4, m)

Andrew Ng

Gradient descent with softmax



Backprop: $dz^{[L]} = \hat{y} - y$ (4,1)

$\frac{\partial J}{\partial z^{[L]}}$

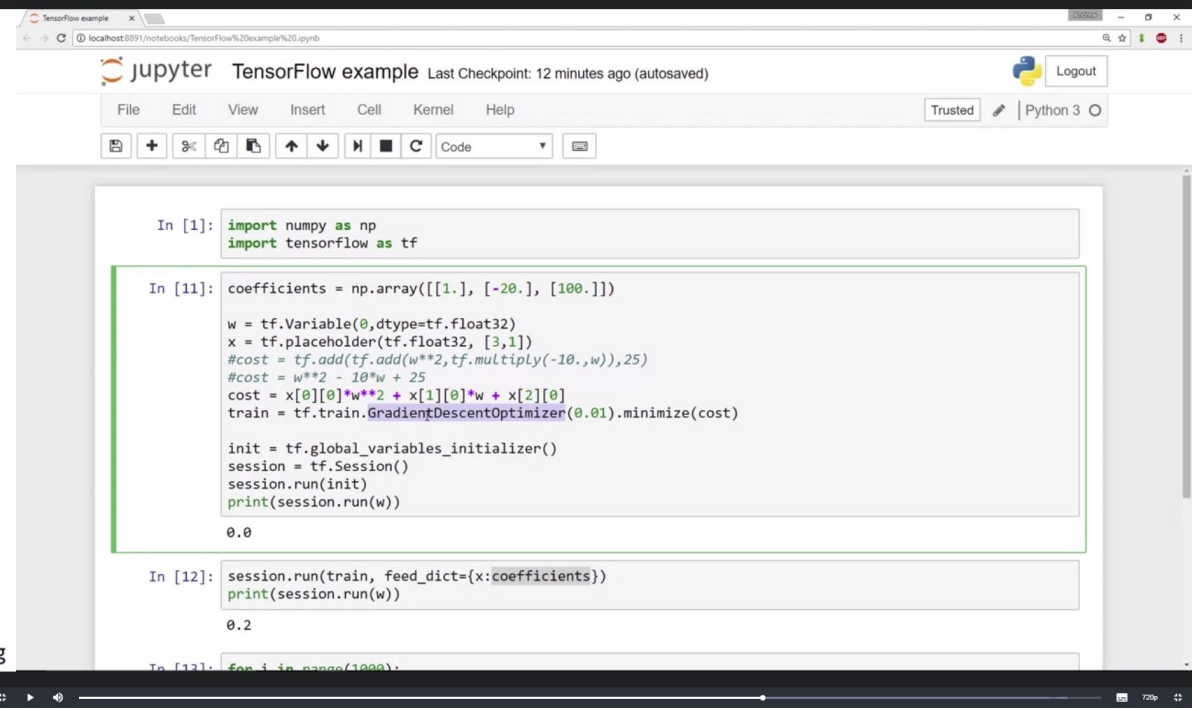
Andrew Ng

Deep learning frameworks

- Caffe/Caffe2
- CNTK
- DL4J
- Keras
- Lasagne
- mxnet
- PaddlePaddle
- TensorFlow
- Theano
- Torch

Choosing deep learning frameworks
- Ease of programming (development and deployment)
- Running speed
- Truly open (open source with good governance)

Andrew Ng



Take aways

- Random hyperparameter search instead of using grids, to cover more variant values for the more important parameter. Then refine coarse to fine.
- Just as normalizing your input, normalize every layer input to alleviate the covariance shift. Also adds small noise in the minibatch – free regularizer.
- Softmax ,a generalization of logistic regression to multiple classes linearizes decision boundaries.
- MatConvNet, the deep learning framework employed in GEOICT lab for the first is not counted. Use Tensorflow to match with fancy stuff.