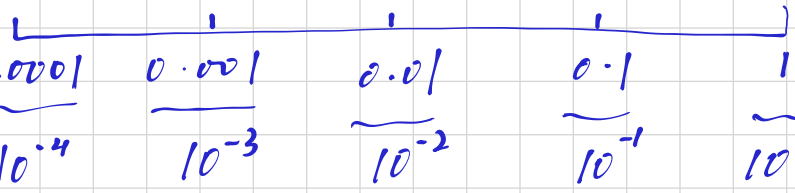
# §Improve DNN - Week 3

## 1. Hyperparameter Tuning

### (1) Appropriate scale for hyperparameters

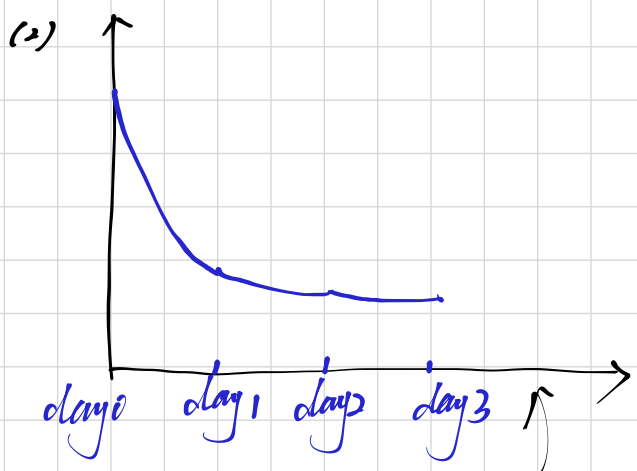



↙ Hyperparameters for exponentially weighted averages

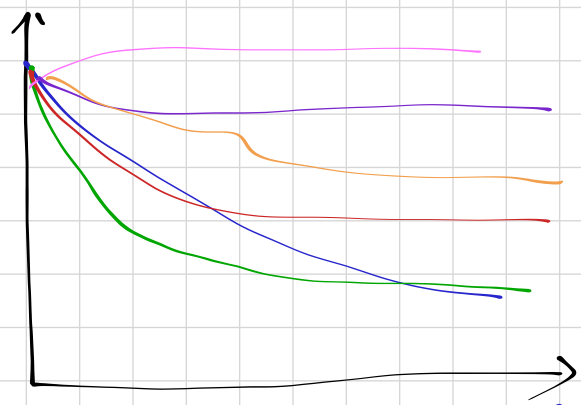
$\Rightarrow \begin{cases} r \in [-3, -1] \\ 1-\beta = 10^{r} \\ \beta = 1-10^{r} \rightarrow \beta = 0.9, 0.99, 0.999 \end{cases}$

↙ Sample more density when $\beta \rightarrow 1$

(2)




Babysitting one model ← Pandas



Caviar

Training many models in parallel

## 2. Batch Normalization

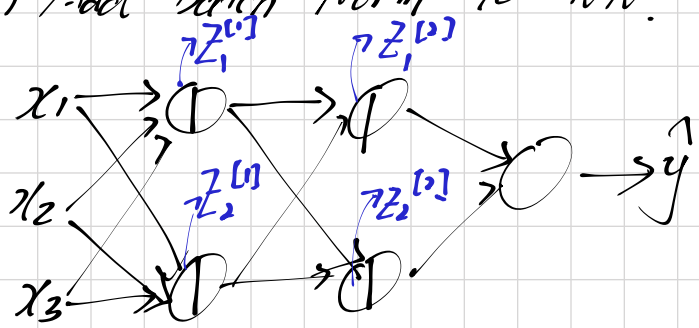
(1) Give some intermediate value in NN: $\underbrace{z^{(1)}, \cdots, z^{(m)}}_{Z^{[l](i)}}$

$$\begin{cases} \mu = \frac{1}{m}\sum_i z^{(i)} \\ \sigma^2 = \frac{1}{m}\sum_i (z_i - \mu)^2 \\ z_{norm}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \\ \tilde{z}^{(i)} = \gamma z_{norm}^{(i)} + \beta \end{cases}$$

If $\gamma = \sqrt{\sigma^2 + \varepsilon}$, $\beta = \mu$, then $\tilde{z}^{(i)} = z^{(i)}$

$\Rightarrow$ use $\tilde{z}^{(i)}$ instead of $z^{(i)}$

(2) Add Batch Norm to NN:




$$X \xrightarrow{W^{[1]}, b^{[1]}} z^{[1]} \xrightarrow[(BN)]{\beta^{[1]}, \gamma^{[1]}} \tilde{z}^{[1]} \rightarrow a^{[1]} = g^{[1]}(\tilde{z}^{[1]}) \rightarrow \cdots$$

$$\cdots \leftarrow a^{[2]} \leftarrow \tilde{z}^{[2]} \xleftarrow[(BN)]{\beta^{[2]}, \gamma^{[2]}} z^{[2]} \leftarrow W^{[2]}, b^{[2]}$$

For mini-batches:

$X^{\{1\}} \xrightarrow{W^{[1]}, b^{[1]}} Z^{[1]} \xrightarrow[\beta^{[1]}, \gamma^{[1]}]{BN} \tilde{Z}^{[1]} \rightarrow g^{[1]}(\tilde{Z}^{[1]}) = A^{[1]} \rightarrow \cdots$

$X^{\{2\}} \rightarrow \cdots$

$X^{\{3\}} \rightarrow \cdots$

parameters: $W^{[l]}, b^{[l]}, \beta^{[l]}, \gamma^{[l]}$

Batch Norm as regularization:

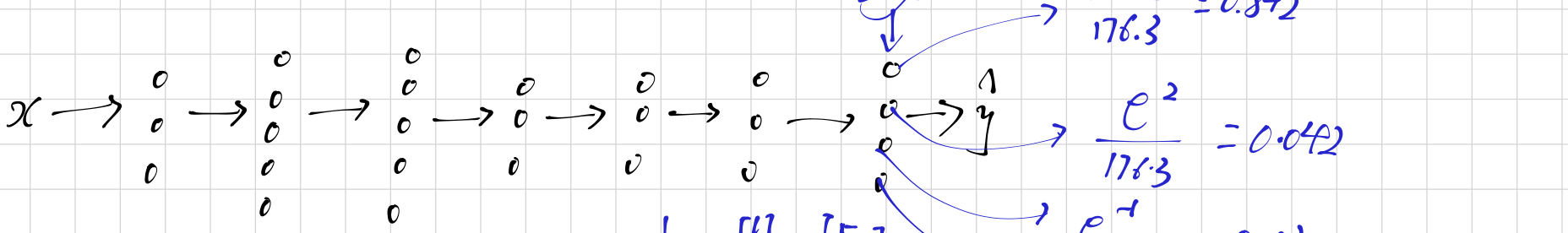
① Each mini-batch is scaled by the mean/variance computed on just that mini-batch.

② This adds some noise to the values $Z^{[l]}$ within that mini batch. So similar to dropout, it adds some noise to each hidden layer's activations.

③ This has a slight regularization effect.

## 3. Multi-class classification

### (1) Softmax Regression




$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$ (4×1)

$t = e^{(z^{[l]})}$ ← (4×1)

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

(4×1)

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

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

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

Softmax regression generalizes logistic regression to C classes.