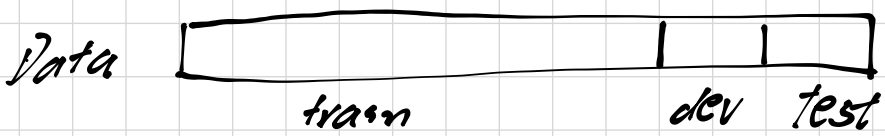
§ Improve deep neural network — week 1

1. Set up ML Application

(1) Train / Dev / Test set



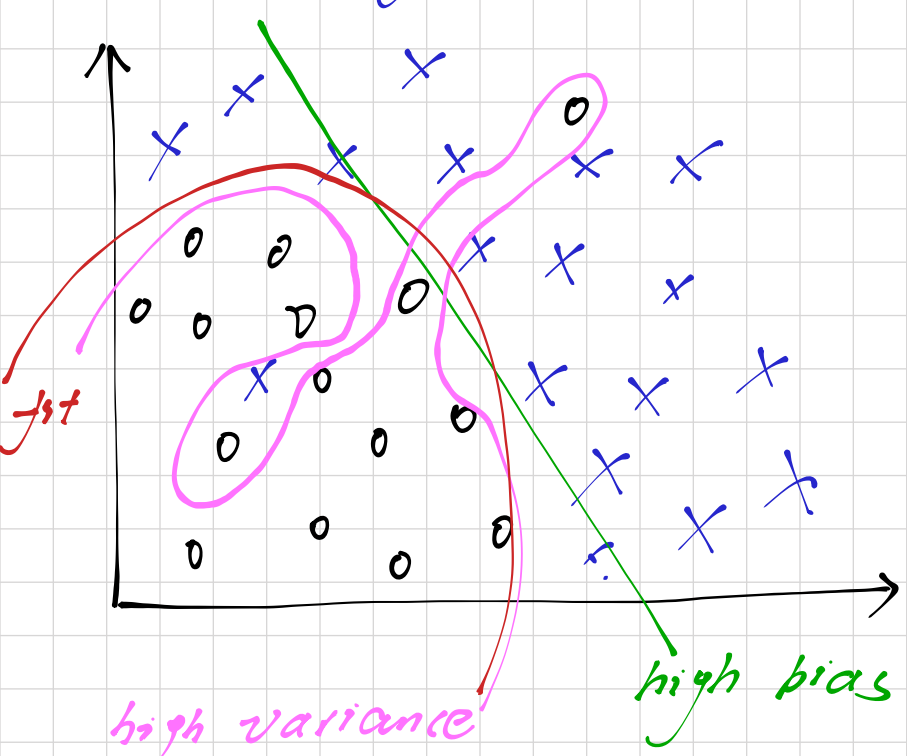
Big Data → eg. 99% / 0.5% / 0.5%

enough to judge algorithm

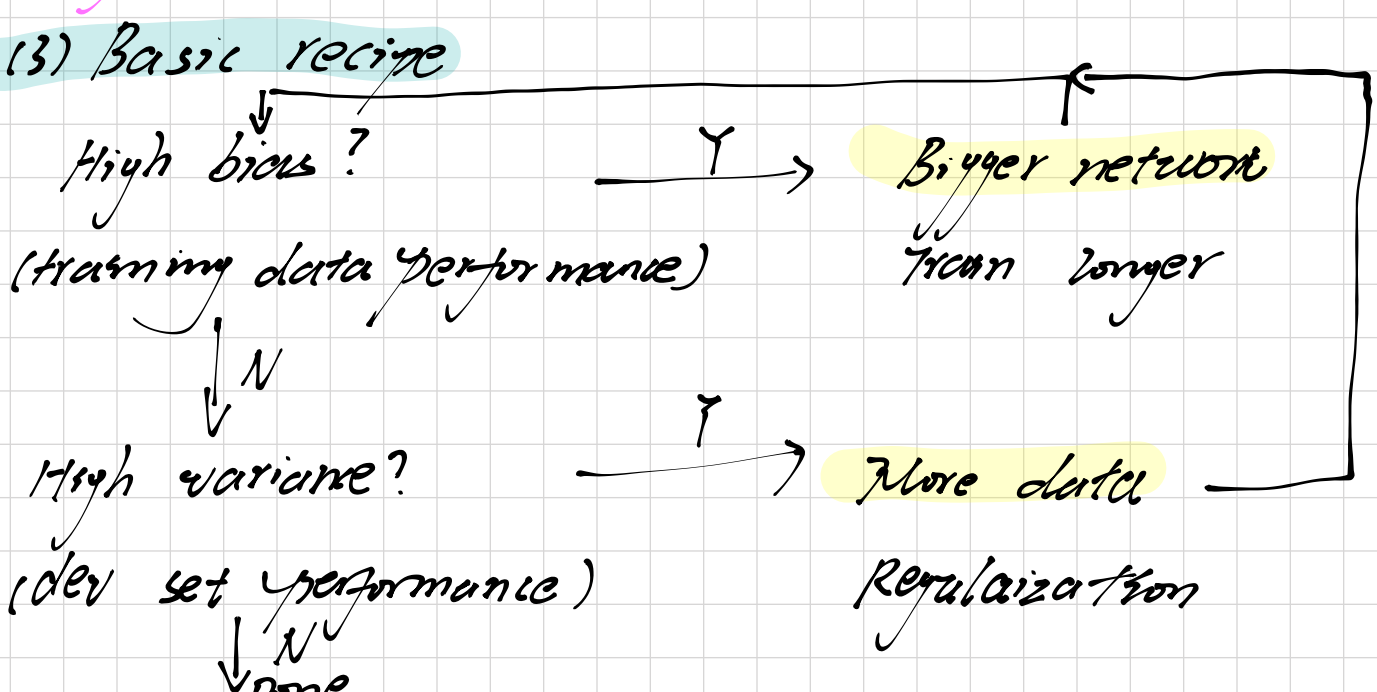
train / test set should match → come from same distribution

(2): Bias / Variance

Train set error:	1%	15%	15%	0.5%
Dev set error:	11%	16%	30%	1%
	high variance	high bias	high variance & high bias	low bias & low variance



(3) Basic recipe



2. Regularizing NN

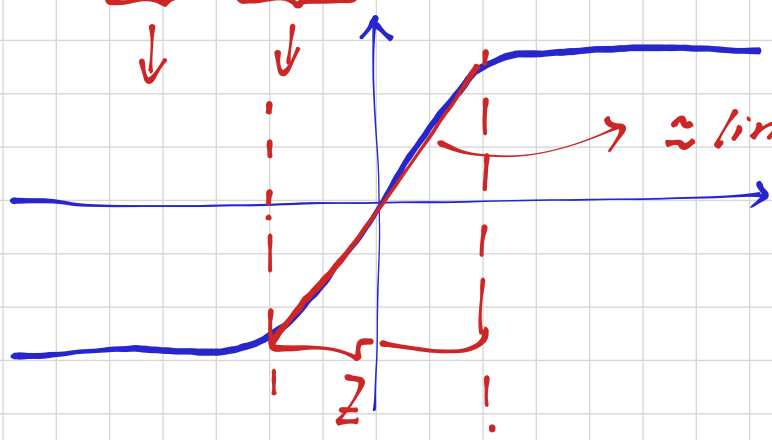
(1) Regularization

$$J(w^{[1]}, b^{[1]}, \dots, w^{[L]}, b^{[L]}) = \frac{1}{m}\sum_{i=1}^{n} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{i=1}^{L} \|w^{[L]}\|_F^2$$

$$\|w^{[L]}\|_F^2 = \sum_{i=1}^{n^{[L-1]}} \sum_{j=1}^{n^{[L]}} (w_{ij}^{[L]})^2$$

↓ prevent overfitting

$\lambda \uparrow \rightarrow w^{[L]} \downarrow \rightarrow z^{[L]} = w^{[L]} a^{[L-1]} + b^{[L]}$



∵ complex → linear

↓

↓ overfitting

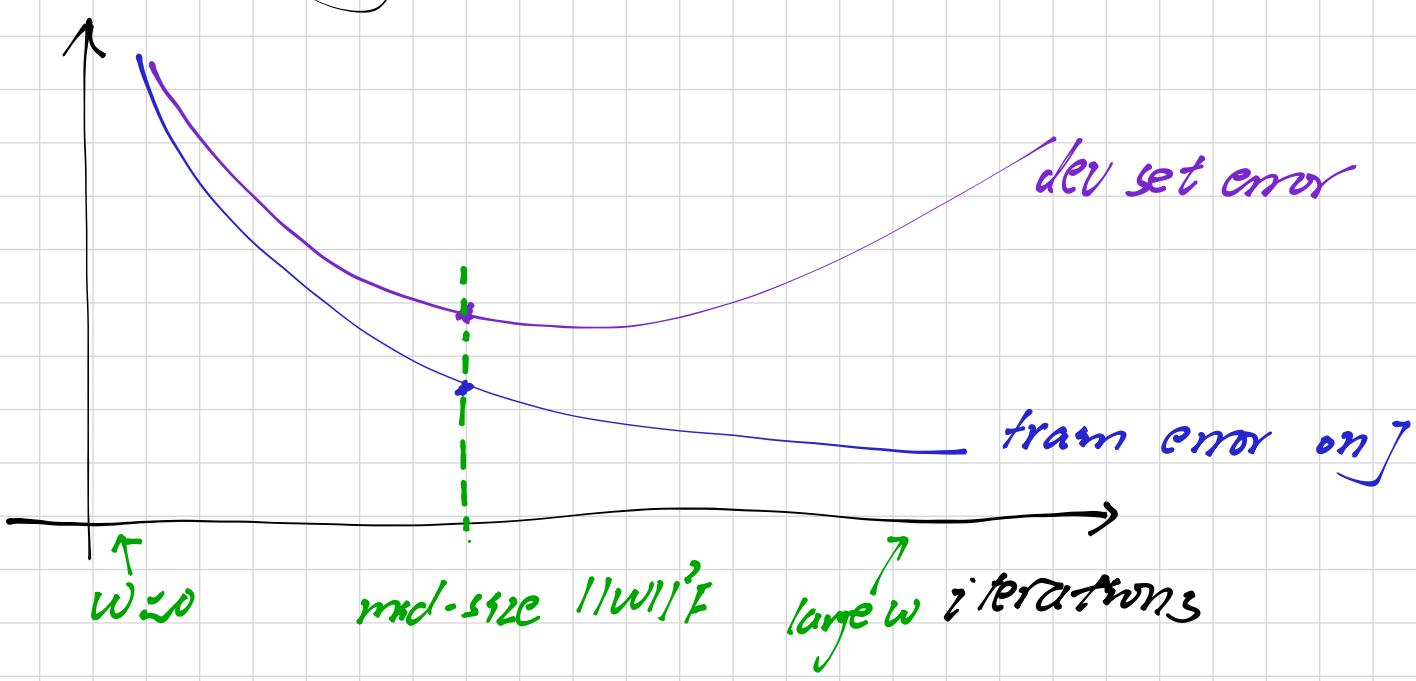
(2) Dropout

① can't rely on any one feature, so have to spread out weights

② choose different keep-prob (depend on the complexity of each layer)

shrink weights

(3) Early stopping



3. Gradient descent check

-for each i:

$$d\theta_{approximate}[i] = \frac{J(\theta_1, \theta_2, \dots, \theta_i + \varepsilon, \dots) - J(\theta_1, \theta_2, \dots, \theta_i - \varepsilon, \dots)}{2\varepsilon}$$

$$\approx d\theta[i] = \frac{\partial J}{\partial \theta_i}$$

check $\frac{\|d\theta_{approximate} - d\theta\|_2}{\|d\theta_{approximate}\|_2 + \|d\theta\|_2} \approx 10^{-7}$ → great

10^{-5}

10^{-3} → worry

① don't use in training — only to debug

② If algorithm fails grad check, look at components to try to identify bug

③ Remember regularization

④ Don't work with dropout

⑤ Run at random initialization; perhaps again after some training