PART 2

PART 1: classic supervised learning

$$\{\bar{x}_i, y_i\}_{i=1}^{N} \Longrightarrow P(Y|\bar{x})$$

PART 2: Deep Supervised learning

Feature extractor | classifier

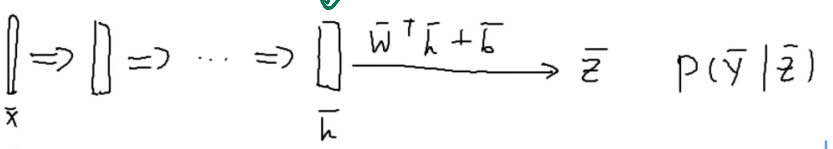
$$\bar{x} \Rightarrow [\,] \Rightarrow [\,] \Rightarrow \cdots \Rightarrow [\,] \Longrightarrow P(Y|h)$$

$$f(x) = h$$

L06: Feedforward NN (FNN)

L07: Convolutional NN (CNN)

penultimate feature layer ↓ logits ↓



$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} p(y_i|x_i;\theta)$$

① $\bar{y}$: real-valued vector

identity matrix ↙

$$p(\bar{y}|\bar{x}) = \mathcal{N}(\bar{y}|\bar{z}, I)$$

per-sample cross entropy loss

$$L = -\log p(\bar{y}|\bar{x},\theta) \propto \frac{1}{2}\|\bar{y}-\bar{z}\|_2^2$$

$$= \frac{1}{2}\left[(y_1-z_1)^2 + (y_2-z_2)^2 + \cdots\right]$$

$$\frac{\partial L}{\partial z_k} = z_k - y_k$$

Error

$\frac{\partial L}{\partial z_2}$

↑ predicted ↑ observed

② $y \in \{0, 1\}$ $\quad p(y|\bar{x}) = \text{Ber}(y|\sigma(z))$

$$p(y=1|\bar{x}) = \sigma(z)$$

$$L = -\log p(y|\bar{x}, \theta) = -[y \log \sigma(z) + (1-y)\log(1-\sigma(z))]$$

predicted ↙ ↘ observed

$$\frac{\partial L}{\partial z} = \sigma(z) - y$$

Error

Sigmoid ok as output Unit

$\sigma(z) - y \approx 0$ {

$y=1, \quad \sigma(z) \approx 1 \quad z >> 0$

Model correct

$y=0, \quad \sigma(z) \approx 0, \quad z << 0$

Model correct

(3) $Y \in \{1, 2, \ldots, C\}$ $\quad \bar{z} = (z_1, z_2, \ldots z_C)^T$

$$p(Y=c \mid \bar{x}) = \frac{e^{z_c}}{\sum_c e^{z_c}}$$

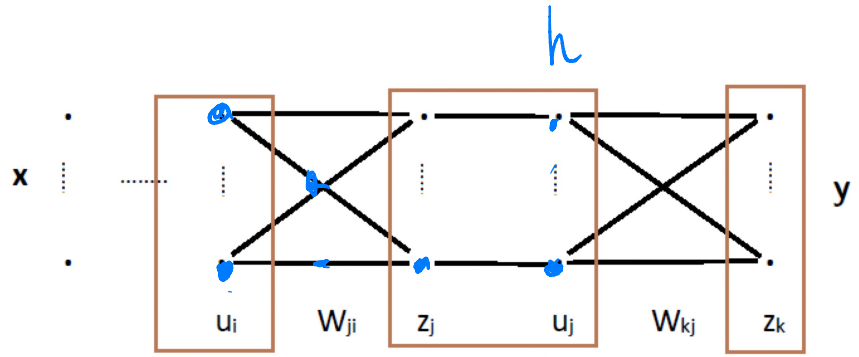
$$L = -\log p(Y \mid \bar{x}, \theta)$$

$$\frac{\partial L}{\partial z_c} = p(Y=c \mid \bar{x}) - \mathbb{1}(Y=c)$$

Error

↑ predicted

↑ observed

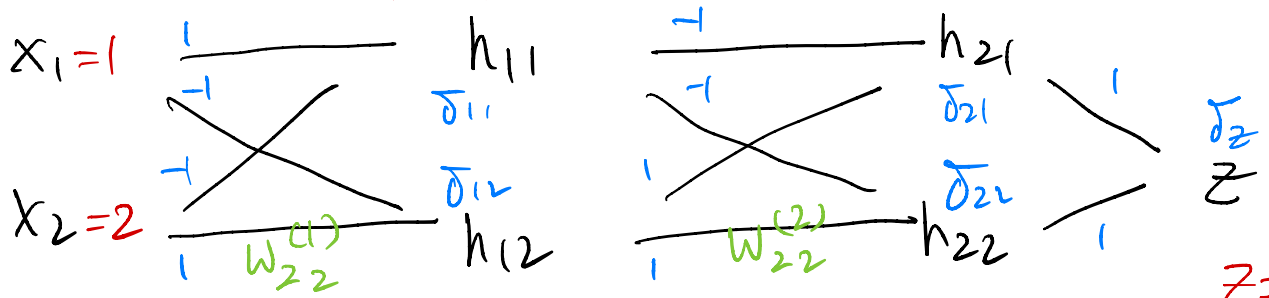


$$z_j = \sum_i u_i W_{ji} + \cdots \quad u_j = g(z_j) \quad z_k = \sum_j u_j W_{kj} + \cdots$$

x_1	x_2	y
1	2	0

h: ReLU

p(y|z): sigmoid



$z=2$ $\quad p(y=1 \mid x_1=1, x_2=2)$

$= \sigma(2)$

Forward

$\delta_{11} = \frac{\partial u_{11}}{\partial z_{11}} (\delta_{21} \times (-1) + \delta_{22} \times (-1))$

$= 0$

$\delta_{21} = \frac{\partial u_{21}}{\partial z_{21}} \delta_z \cdot 1$

$= 0.88$

$\delta_z = \sigma(2) - y$

≈ 0.88

$\delta_{12} = \frac{\partial u_{12}}{\partial z_{12}} (\delta_{21} \cdot 1 + \delta_{22} \cdot 1)$

$= 1\,(0.88 \times 1 + 0.88 \times 1)$

$= 1.76$

$\delta_{22} = 0.88$

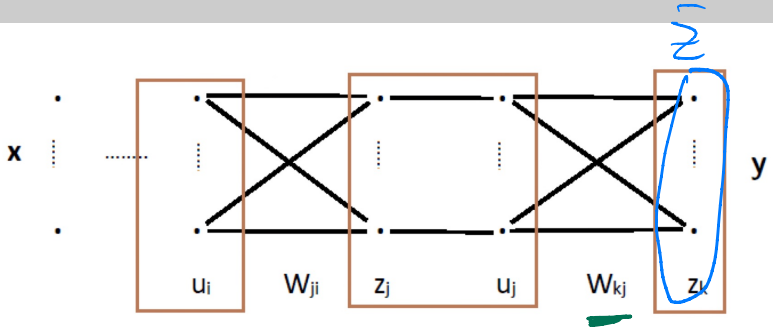
Backward

$\frac{\partial L}{\partial w_{22}^{(1)}} = \underset{1 \times 1.76}{\text{input}} \times \underset{\text{error}}{\frac{\partial L}{\partial w_{12}^{(2)}}} = \_\_\_$

Derivation of Back prop

$L = -\log p(y \mid \bar{x}, \theta)$

$= -\log p(y \mid \bar{z}, \theta)$



$$z_k = \sum_j u_j W_{kj}, \quad u_j = g(z_j), \quad z_j = \sum_i u_i W_{ji}$$

$$\frac{\partial L}{\partial W_{kj}} = \frac{\partial L}{\partial z_k} \frac{\partial z_k}{\partial W_{kj}}$$

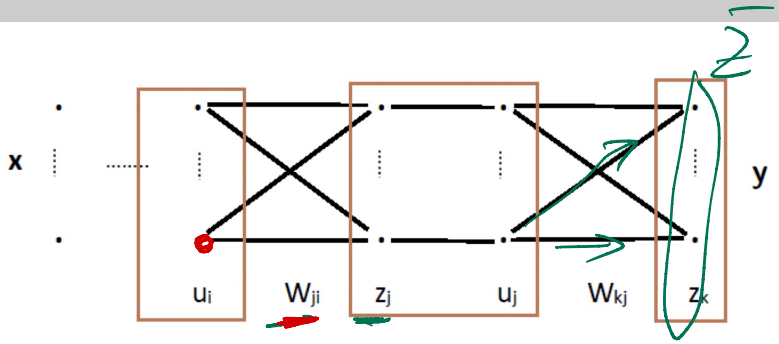
$W_{kj} \rightarrow z_k \rightarrow L$

$$= \delta_k \; u_j$$

$$= u_j \; \delta_k$$

input error

$W_{ji} \to z_j \to u_j \to \{z_k\} \to L$



$$z_k = \sum_j u_j W_{kj}, \quad u_j = g(z_j), \quad z_j = \sum_i u_i W_{ji}$$

$$\frac{\partial L}{\partial W_{ji}} = \sum_k \frac{\partial L}{\partial z_k} \frac{\partial z_k}{\partial W_{ji}}$$

$$= \sum_k \delta_k \frac{\partial z_k}{\partial u_j} \frac{\partial u_j}{\partial z_j} \frac{\partial z_j}{\partial W_{ji}}$$

$$= \sum_k \delta_k W_{kj} \frac{\partial u_j}{\partial z_j} u_i$$

$$= u_i \frac{\partial u_j}{\partial z_j} \sum_k W_{kj} \delta_k$$

u_i: input

$\sum_k W_{kj} \delta_k$: δ_j

$z_k = u_{j_1} W_{kj_1} + u_{j_2} W_{kj_2}$

$\frac{\partial z_k}{\partial u_{j_1}} = W_{kj_1}$

SGD: $\bar{\theta} \leftarrow \bar{\theta} - \varepsilon \bar{g}$: push $\bar{\theta}$ in direction $-\bar{g}$ for ε distance, stop

Momentum

* $\bar{\theta}$ has speed $\bar{v}$
* At each iteration $0 < \alpha < 1$
  * Current speed reduce $\bar{v} \leftarrow \alpha \bar{v}$
  * Get acceleration $-\varepsilon \bar{g}$
  * New speed: $\bar{v} \leftarrow \alpha \bar{v} - \varepsilon \bar{g}$
  * $\bar{\theta} \leftarrow \bar{\theta} + \bar{v}$

t	V
0	$\bar{V}_0 = 0$
1	$\bar{V}_1 = \alpha \bar{V}_0 - \varepsilon \bar{g}_1 = -\varepsilon \bar{g}_1$
2	$\bar{V}_2 = \alpha \bar{V}_1 - \varepsilon \bar{g}_2 = -\varepsilon(\alpha \bar{g}_1 + \bar{g}_2)$
3	$\bar{V}_3 = \alpha \bar{V}_2 - \varepsilon \bar{g}_3 = -\varepsilon(\alpha^2 \bar{g}_1 + \alpha \bar{g}_2 + \bar{g}_3)$

$0 < \alpha < 1$

Exponentially decaying average of past gradients

Ada Grad (Adaptive Gradient)

$\bar{\theta} = (\theta_1, \theta_2, \ldots \theta_{1,000,000})$

$\bar{g} = (g_1, g_2, \ldots g_{1,000,000})$

Past change (gradient)

$\bar{r} = (r_1, r_2, \ldots r_{1,000,00})$

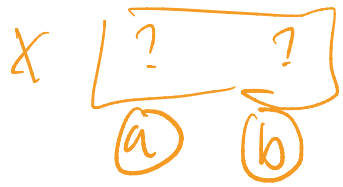
$\bar{r} = 0$

$\bar{r} \leftarrow \bar{r} + \bar{g} \odot \bar{g}$ $\quad \bar{g}^2$

↑ Componentwise product

After t iterations: $\bar{r} = \bar{g}_1^2 + \bar{g}_2^2 + \ldots + \bar{g}_t^2$

Para's that have changed a lot already should change less in future



$\bar{g}_1 = (g_{11}, g_{12} \ldots g_{1,000,00})$

$\bar{g}_2 = (g_{21}, g_{22}, \ldots)$

SGD: $\theta \leftarrow \theta - \varepsilon \bar{g}$ $\quad \Delta\theta = -\varepsilon \bar{g}$

AdaGrad: $\Delta\theta = -\frac{1}{\delta + \sqrt{\bar{r}}} \odot \varepsilon \bar{g}$

δ: Smoothing constant

$\bar{r} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$ 1,000,000

$\bar{g} = \begin{bmatrix} 0 \\ \\ \\ \end{bmatrix}$

RMS prop (Root mean square prop)

$$\bar{r} \leftarrow \rho \bar{r} + (1-\rho) g^2$$

$0 < \rho < 1$

AdaGrad: $\bar{r} \leftarrow \bar{r} + g^2$

After t iteration:

$$r_t = (1-\rho)(\rho^{t-1} g_1^2 + \rho^{t-2} g_2^2 + \cdots + g_{t-1}^2)$$

Momentum

$$\bar{v} \leftarrow \alpha \bar{v} - \varepsilon \bar{g} \qquad \Delta\bar{\theta} = \bar{v}$$

$$\bar{u} = -\bar{v}$$

$$\bar{u} \leftarrow \alpha \bar{u} + \varepsilon \bar{g} \qquad \Delta\bar{\theta} = -\bar{u}$$

RMS prop

$$\bar{r} \leftarrow \rho \bar{r} + (1-\rho)\, \bar{g}^2$$

$$\Delta\bar{\theta} = -\varepsilon \bar{g} \frac{1}{\delta + \sqrt{\bar{r}}}$$

Adam (Adaptive Moments)

$$\bar{s} \leftarrow \rho_1 \bar{s} + (1-\rho_1)\, \bar{g}$$

$$\bar{r} \leftarrow \rho_2 \bar{r} + (1-\rho_2)\, \bar{g}^2$$

$$\bar{s} \leftarrow \frac{1}{1-\rho_1^t} \bar{s}$$

$$\bar{r} \leftarrow \frac{1}{1-\rho_2^t} \bar{r}$$

moment correction

$$\Delta\theta = -\varepsilon \bar{s} \oslash \frac{1}{\delta + \sqrt{\bar{r}}}$$

$$S \leftarrow \rho_1 \bar{S} + (1-\rho_1)\bar{g}$$

$$\hat{S} = \frac{1}{1-\rho_1 t} S$$

t	S
1	$(1-\rho_1) g_1$
2	$(1-\rho_1)(\rho_1 g_1 + g_2)$
3	$(1-\rho_1)(\rho_1^2 g_1 + \rho_1 g_2 + g_3)$
⋮	

$$\frac{(1-\rho_1)(\rho_1^2 g_1 + \rho_1 g_2 + g_3)}{(1-\rho_1)(\rho_1^2 + \rho_1 + 1)} \quad z$$

$$= \frac{\rho_1^2}{z} g_1 + \frac{\rho_1}{z} g_2 + \frac{1}{z} g_3$$

$$\textcircled{+} = 1$$

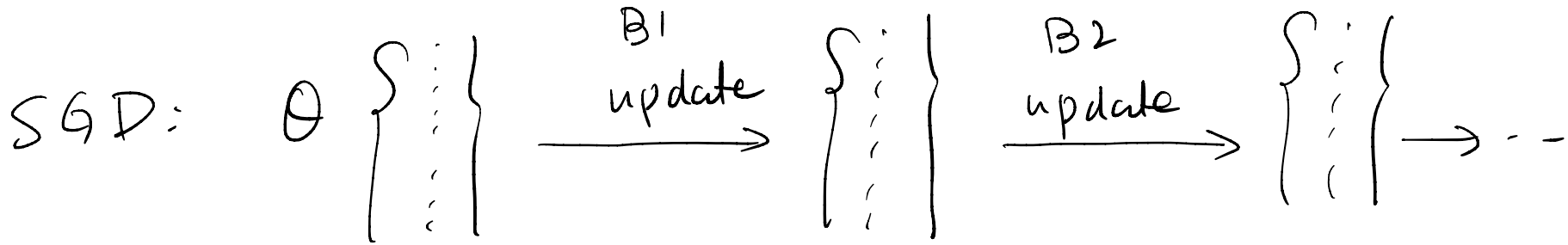
$$p(2)$$

$$E[g_t] = \int p(t)\, g_t\, dt$$

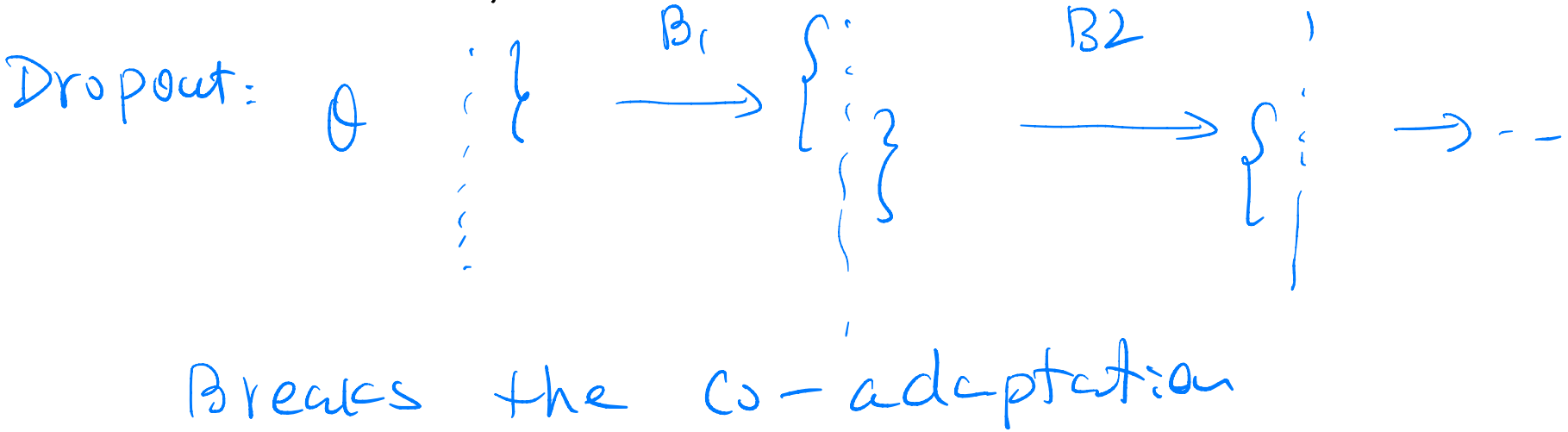
1st order moment

r: $\int p(t)\, g_t^2\, dt$

2nd order moment

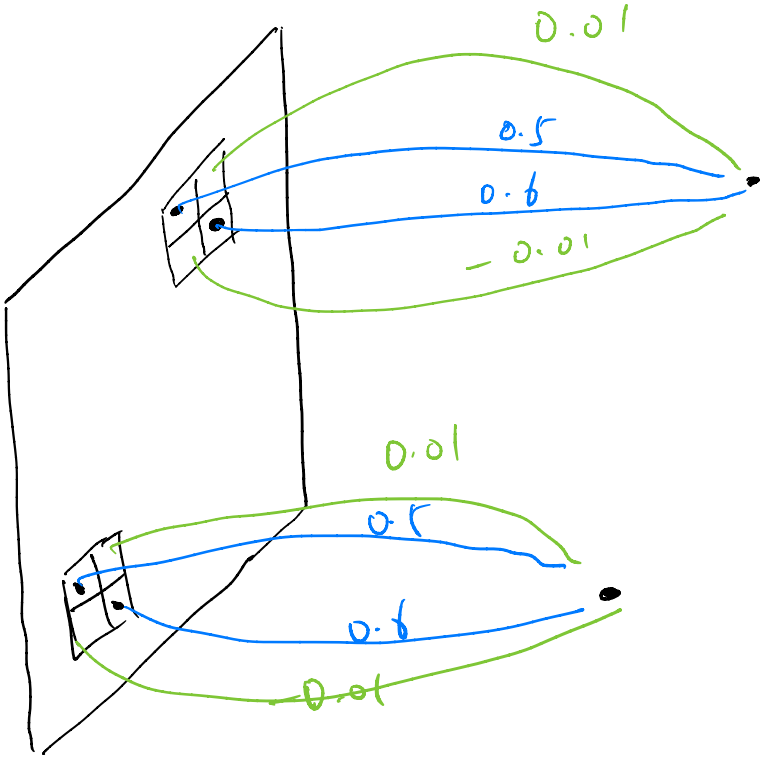


All parameters can co-adapt to fit noise



Breaks the co-adaptation

L07 : CNN



* one unit for detecting one feature

* Detect the same feature else where

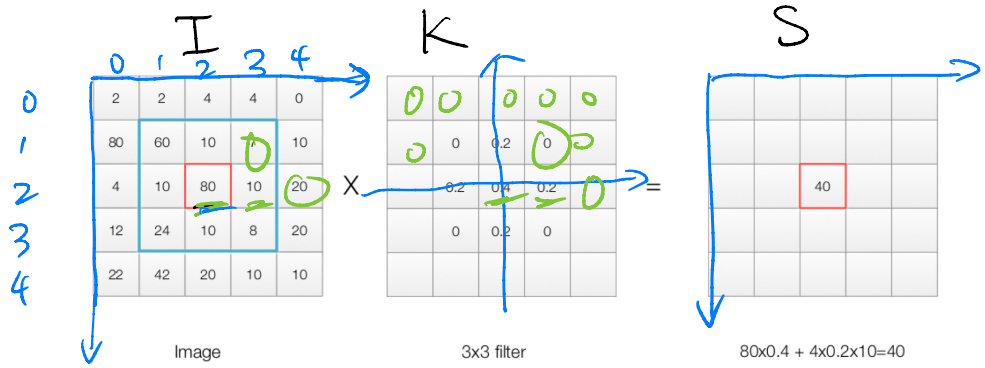
Share parameter

Another inductive bias:
Shift invariant

Filter / Kernel:

0.5	0.01
-0.01	0.6

* Detect multiple feature: Multiple filters



$$S(2,2) = I(2,2)\,K(0,0) + I(3,2)\,K(1,0) + \cdots$$

$$S(i,j) = \sum_{m,n} I(i+m, j+n)\,K(m,n)$$

Cross-correlation

$$K'(m,n) = K(-m,-n)$$

$$\Rightarrow K(m,n) = K'(-m,-n)$$

$$S(i,j) = \sum_{m,n} I(i+m, j+n)\, K(m,n)$$

$$= \sum_{m,n} I(i+m, j+n)\, K'(-m,-n)$$

$a = -m$

$b = -n$

$$= \sum_{a,b} I(i-a, j-b)\, K'(a,b)$$

Convolution

Input : $W_1 \times H_1 \times D_1$

K FxF filters

stride : S

padding : P

Output: $W_2 \times H_2 \times D_2$

$D_2 = K$

$W_2 = \frac{1}{S}(W_1 + 2P - F) + 1$

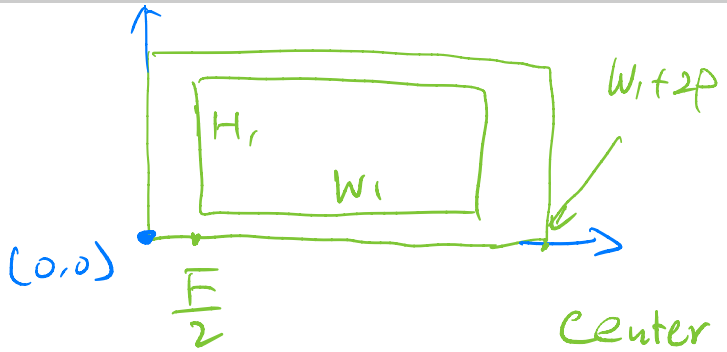
$H_2 = \frac{1}{S}(H_1 + 2P - F) + 1$

# of paras :

$(F^2 + 1)K$

# of FLOPS $= 2 F^2 D_1 W_2 H_2 D_2$

Floating point operations



on X-axis:

center of filter

1st possible location for filter

$\frac{F}{2}$

last possible location ---

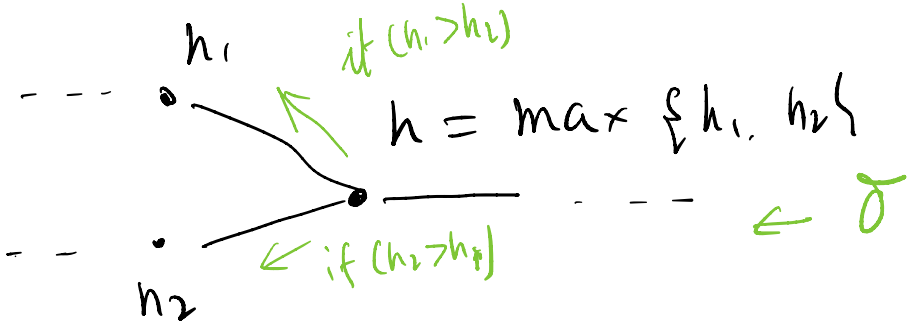
$W_1 + 2P - \frac{F}{2}$

# of possible location ---

$$\frac{W_1 + 2P - \frac{F}{2} - \frac{F}{2}}{S} + 1$$

$= \frac{1}{S}(W_1 + 2P - F) + 1$

Max Pooling : Back prog



Switch

$$\frac{\partial h}{\partial h_1} = \begin{cases} 1 & h_1 > h_2 \\ 0 & h_1 < h_2 \end{cases}$$

$$\frac{\partial h}{\partial h_2} = \begin{cases} 1 & h_2 > h_1 \\ 0 & h_2 < h_1 \end{cases}$$

Data Normalization

$x_1 \in [0, 1]$, $x_2 \in [10, 1{,}000]$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

$$\Rightarrow \quad J(\bar{w}) = E[L(y, \hat{y})] + \lambda (w_1^2 + w_2^2)$$

Regularization affects which parameter more: w_1 w_2

	w_1^2 ↘	w_2^2 ↘
impact on $\hat{y}$	less	more
impact on L:	less	more

Ridge regression: * Reduce $w_1^2 + w_2^2$
* Minimize L

What to do: Reduce w_1^2 more $\Rightarrow$ bias

$x_1 \in [0, 1]$, $x_2 \in [10, 1{,}000]$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

$$J(\bar{w}) = E[L(y, \hat{y})] + \lambda (w_1^2 + w_2^2)$$

$$\frac{\partial J}{\partial w_1} = E\left[\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_1}\right] + 2\lambda w_1$$

$$= E\left[\frac{\partial L}{\partial \hat{y}}\right] x_1 + 2\lambda w_1$$

Smaller

$$\frac{\partial J}{\partial w_2} = E\left[\frac{\partial L}{\partial \hat{y}}\right] x_2 + 2\lambda w_2$$

larger

Unnormalized Data

x_{11}	x_{12}	..	x_{1D}
x_{21}	x_{22},	..	x_{2D}
-	-	-	-
x_{N1},	x_{N2}		x_{ND}
μ:	μ_1	μ_2	μ_D
σ:	σ_1	σ_2	.. σ_D

Normalized Data

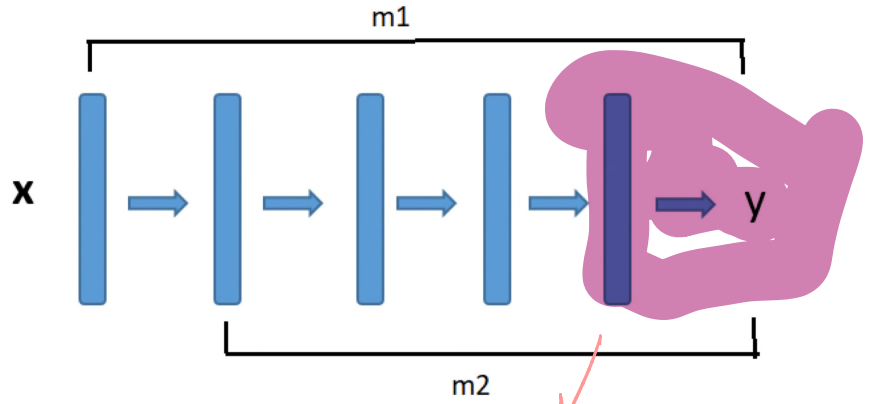
$\hat{x}_{11}$	$\hat{x}_{12}$	- -	$\hat{x}_{1D}$
-	-		-
$\hat{x}_{N1}$	$\hat{x}_{N2}$		$\hat{x}_{ND}$
μ:	0	0	0
σ:	1	1	1

$$\mu_j = \sum_{i=1}^{N} x_{ij} \quad \text{mean of c}$$

$$\sigma_j^2 = \frac{1}{N} \sum_{i=1}^{N} (x_{ij} - \mu_j)^2$$

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_i}$$

Batch Normalization

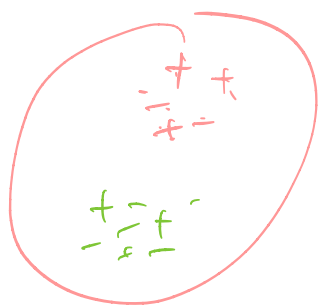


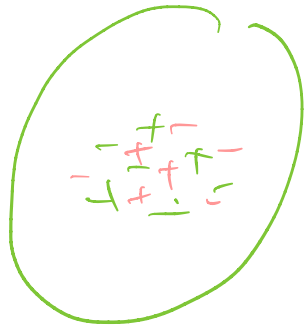
latent space

Without Normalization

iteration 10

iteration 15





Covariate Shift

Inception Module

Input: $H_1 \times W_1 \times D_1$

# of para's

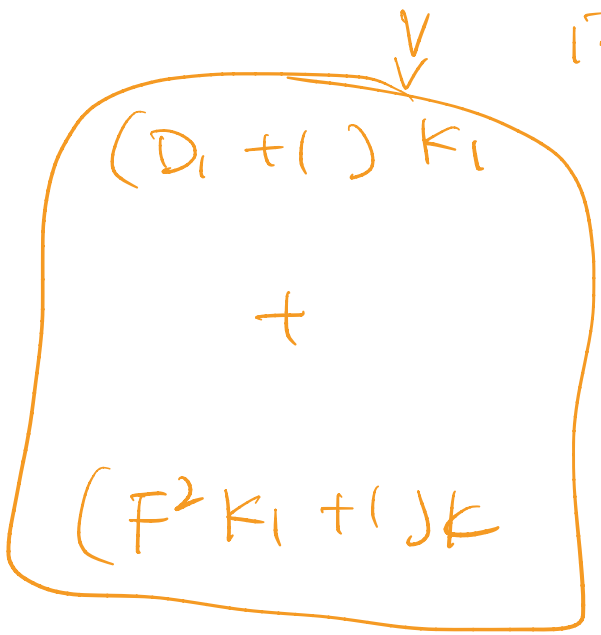
(a) K $F \times F$ filters $\quad (F^2 D_1 + 1) K$

$\Rightarrow$ Output 1

(b) ① K_1 1×1 filters

$\Rightarrow$ intermediate output: $H_1 \times W_1 \times K_1$

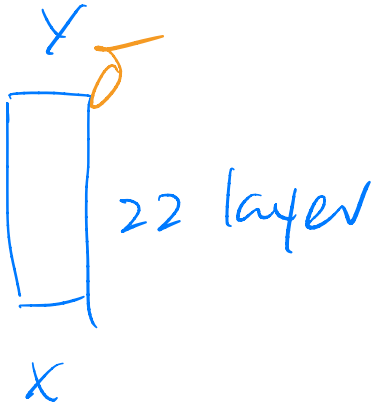
② K $F \times F$ filters

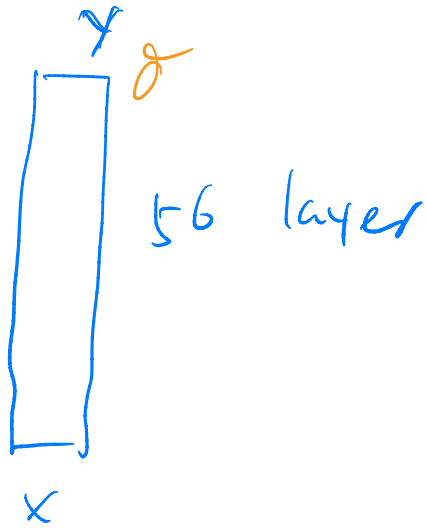


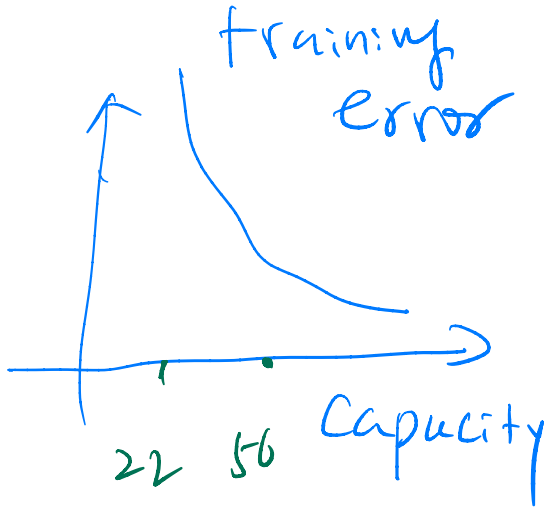
$\Rightarrow$ Output 2:

Same shape as output 1

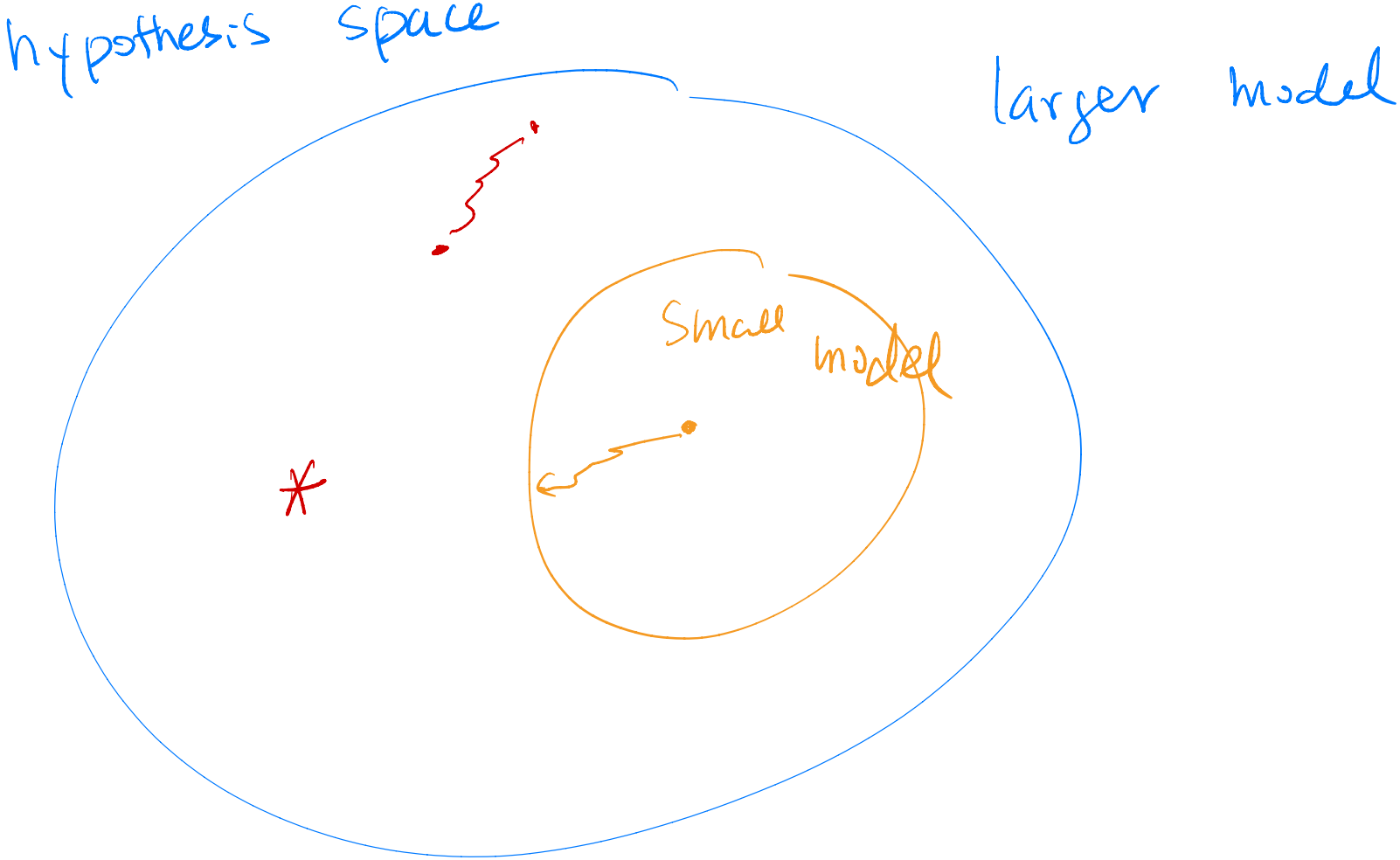
Gradient





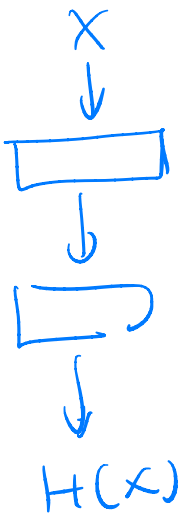


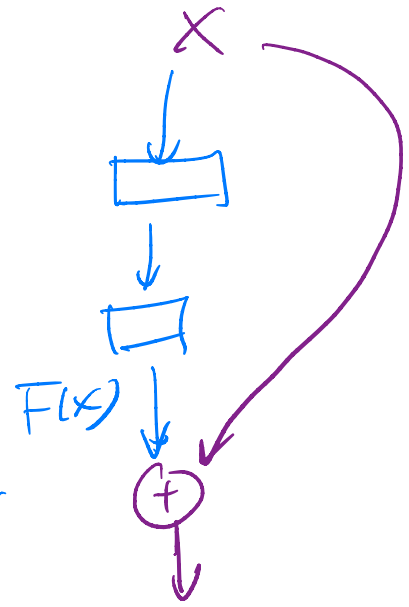
Training error			
Theory	high	lower	layer model
Practice	lower	higher	layer model


hypothesis space
larger model
small model

Residual Module

plain net





residual connection

x identity connection

$$H(x) = F(x) + x$$

$$F(x) = \underbrace{H(x) - x}_{\text{residual}}$$