

# 6주차 발표

BOAZ 17기 이소정





#17

Advanced Optimizer than SGD

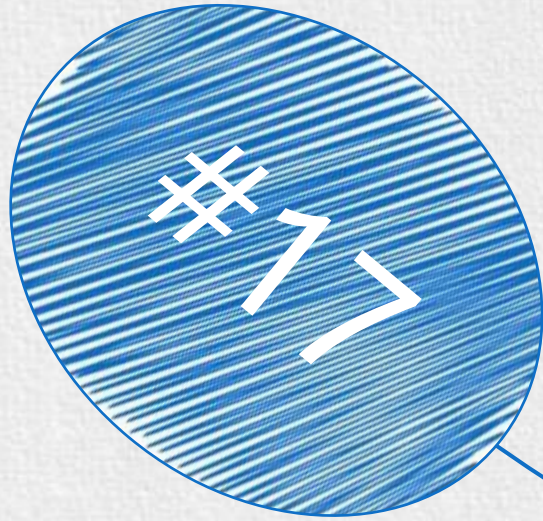
#19

Assignment #3 Review

#20

Basic of Convolutional Neural Network





## Topics to learn today

1. Review from last lecture
2. Batch/Stochastic Gradient Descent
3. Advanced Gradient Descent Algorithms
4. How to visualize the result



A large, irregular rectangular area filled with dense, diagonal blue scribbles, resembling a hand-drawn or painted texture. The scribbles are in various shades of blue and are oriented diagonally from the top-left to the bottom-right.

*Batch/Stochastic Gradient Descent*



#17

## Gradient Descent

$$\theta = \theta - \eta \nabla J(\theta)$$

$\theta$ : parameter set of the model

$\eta$ : Learning rate

$J(\theta)$ : Loss function

#17

## 1. Batch Gradient Descent

- Calculate gradient of parameters for whole training dataset.
- Need a lot of memory depending on data.
- Calculating gradient is too slow, thus optimization is slow.



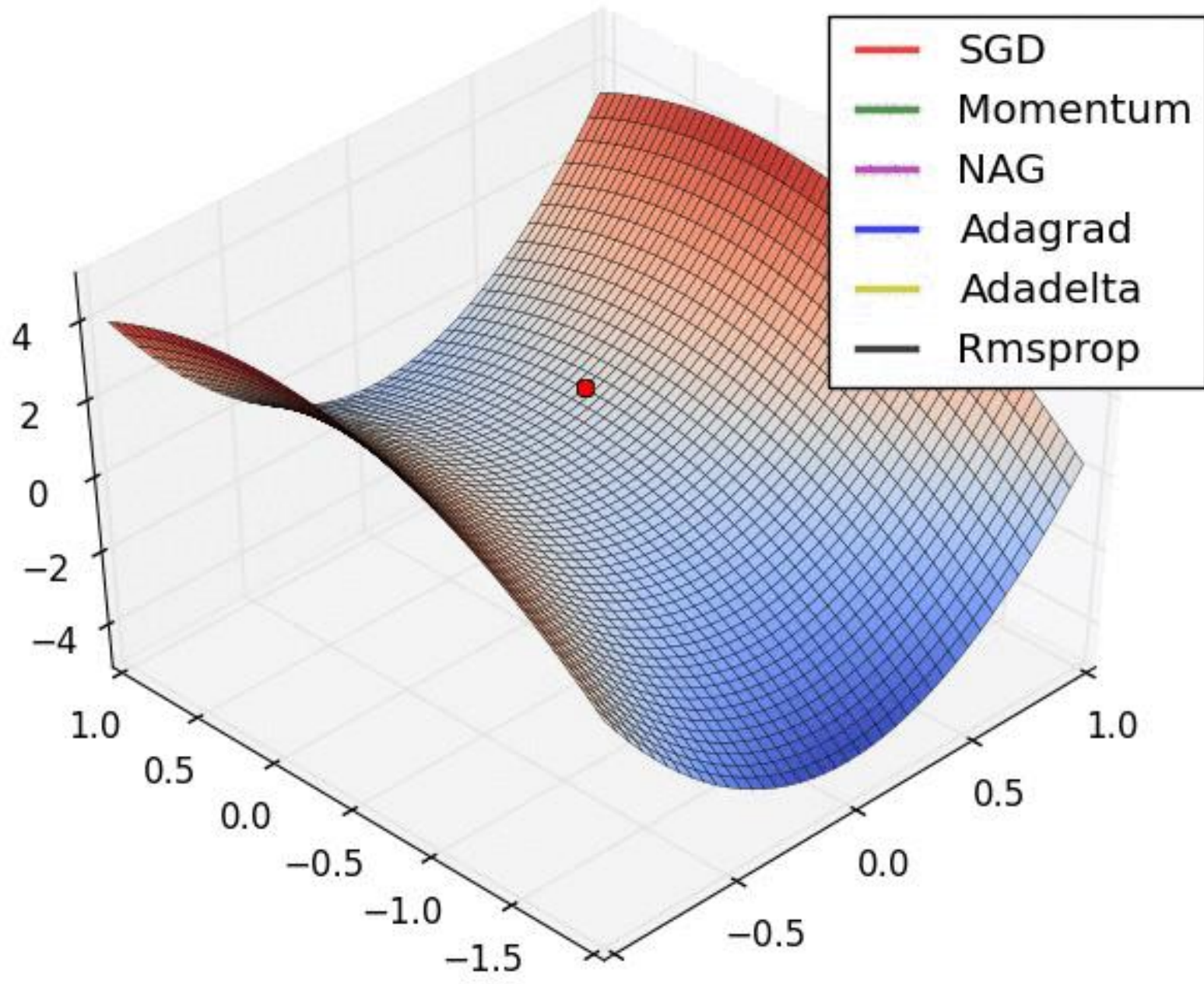
- Calculate gradient for small chunk of whole training dataset (**mini-batch**) rather than the whole training dataset (**batch**).
- Stochastic since the gradient is not deterministic, but stochastic depending on the mini-batch.
- Faster than batch gradient descent, while converging similar.
- Can avoid local minima by stochasticity.

→ Calculating  
dataset  
dataset

→ Stochastic  
stochastic

→ Faster

→ Can a



g

t

g similar.

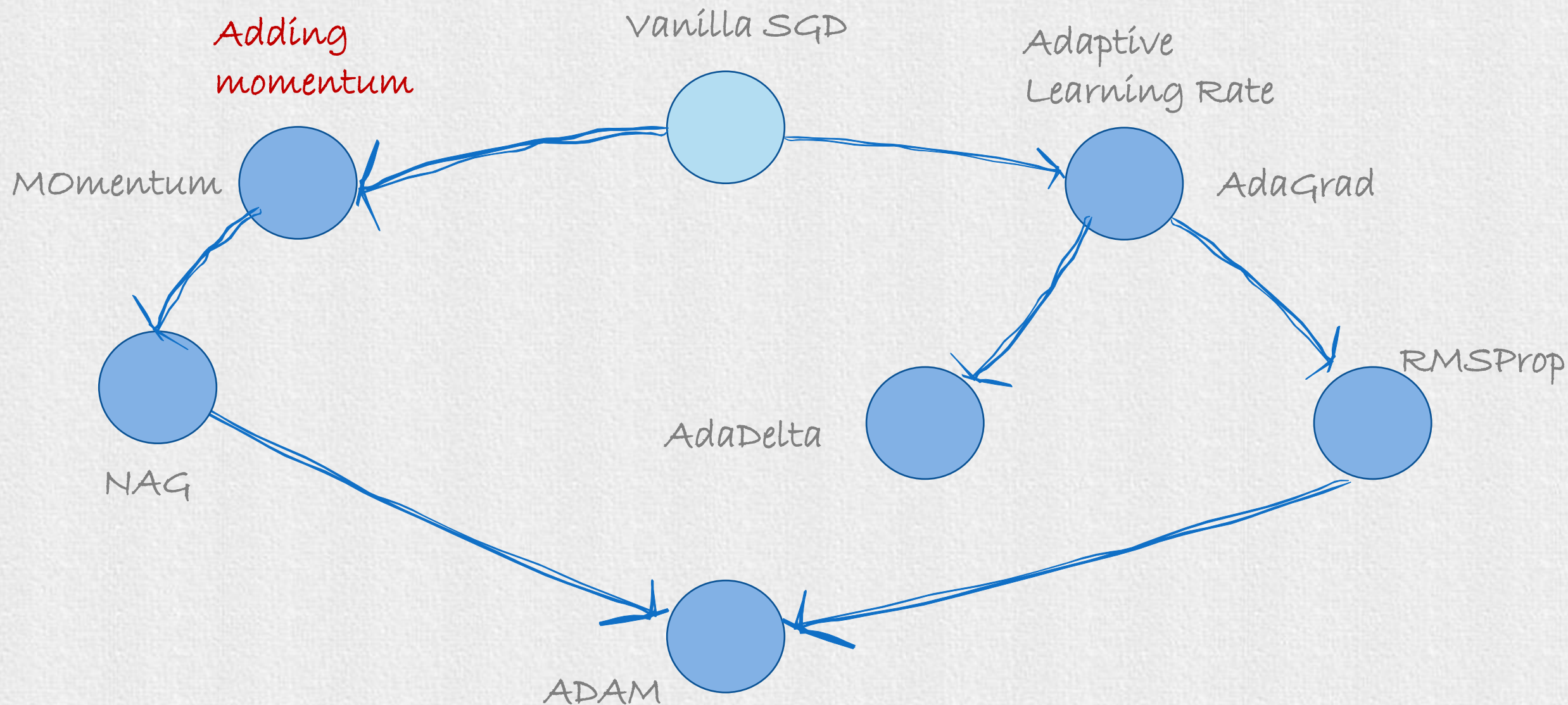


The background of the slide is a light gray with a large, irregular, blue scribbled area in the center. The scribbles are made of many overlapping, diagonal, brushstroke-like lines in various shades of blue, creating a textured, hand-drawn effect.

# *Advanced Gradient Descent Methods*

#17

## Diagram of Gradient Descent Development





#17

## Momentum

$$\theta = \theta - v_t$$

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$



Stochastic Gradient  
Descent **without**  
Momentum

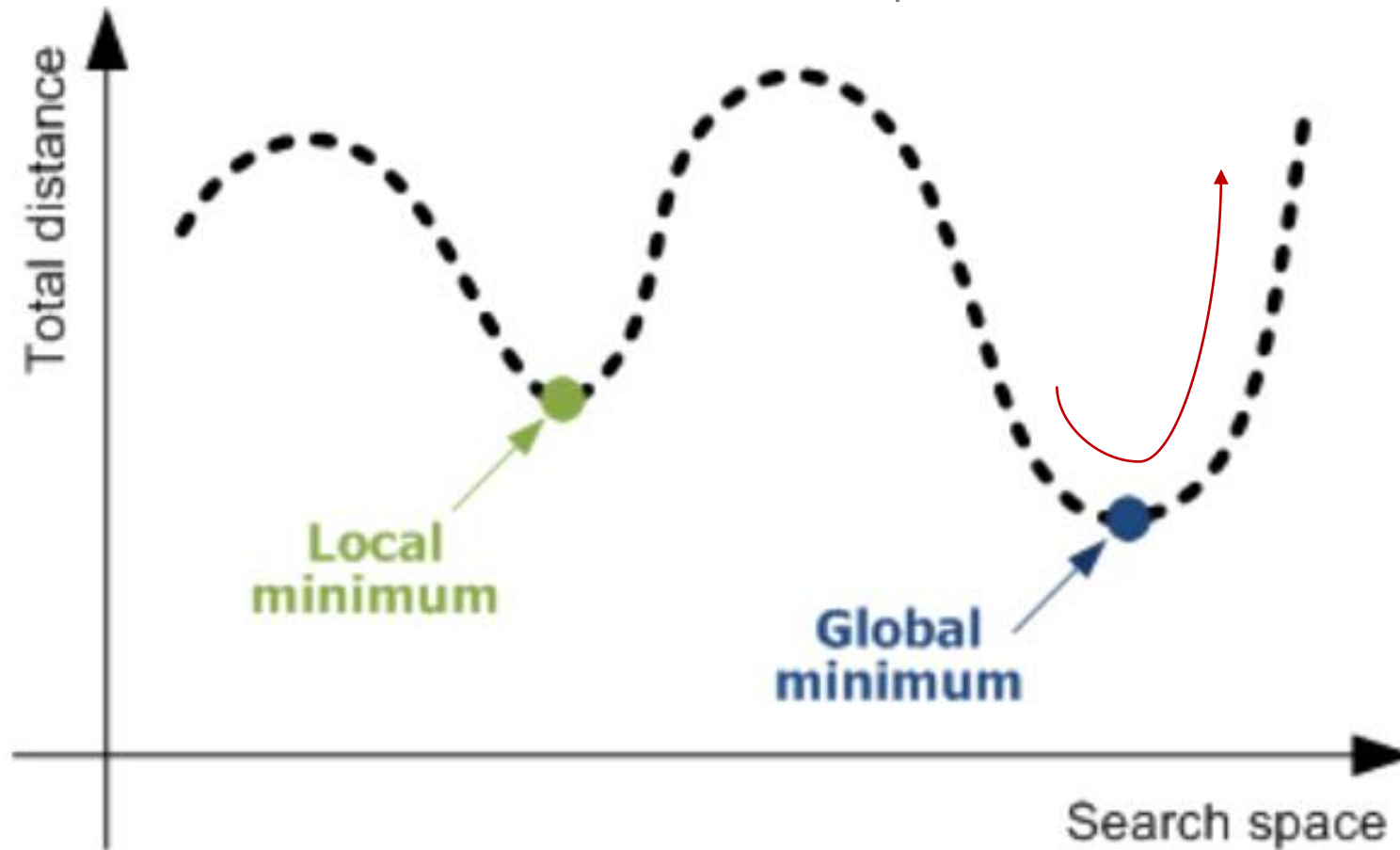


Stochastic Gradient  
Descent **with**  
Momentum

#17

## Momentum

Can escape local minima, but cannot stop or slow at global minima!!





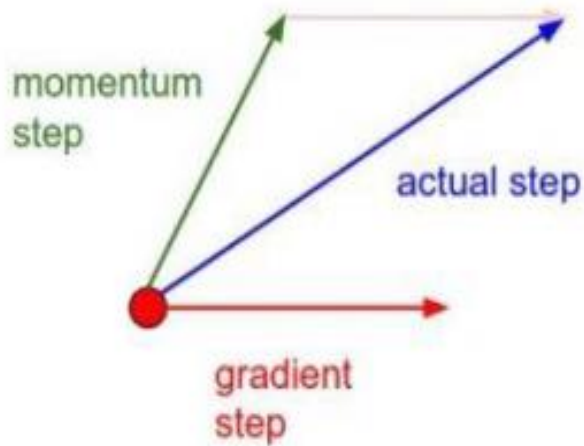
#17

## Nesterov Accelerated Gradient (NAG)

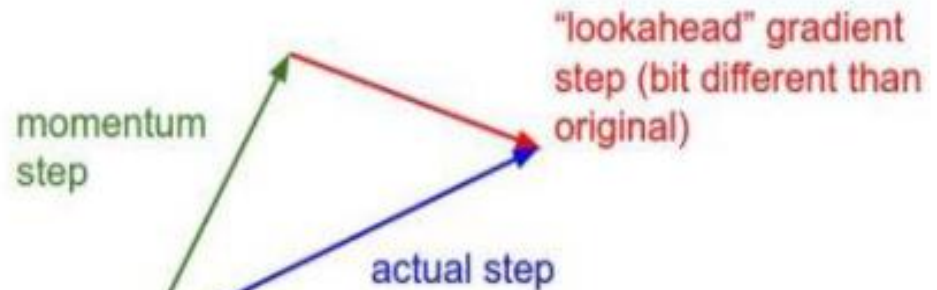
$$\theta = \theta - v_t$$

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta - \gamma v_{t-1})$$

Momentum update

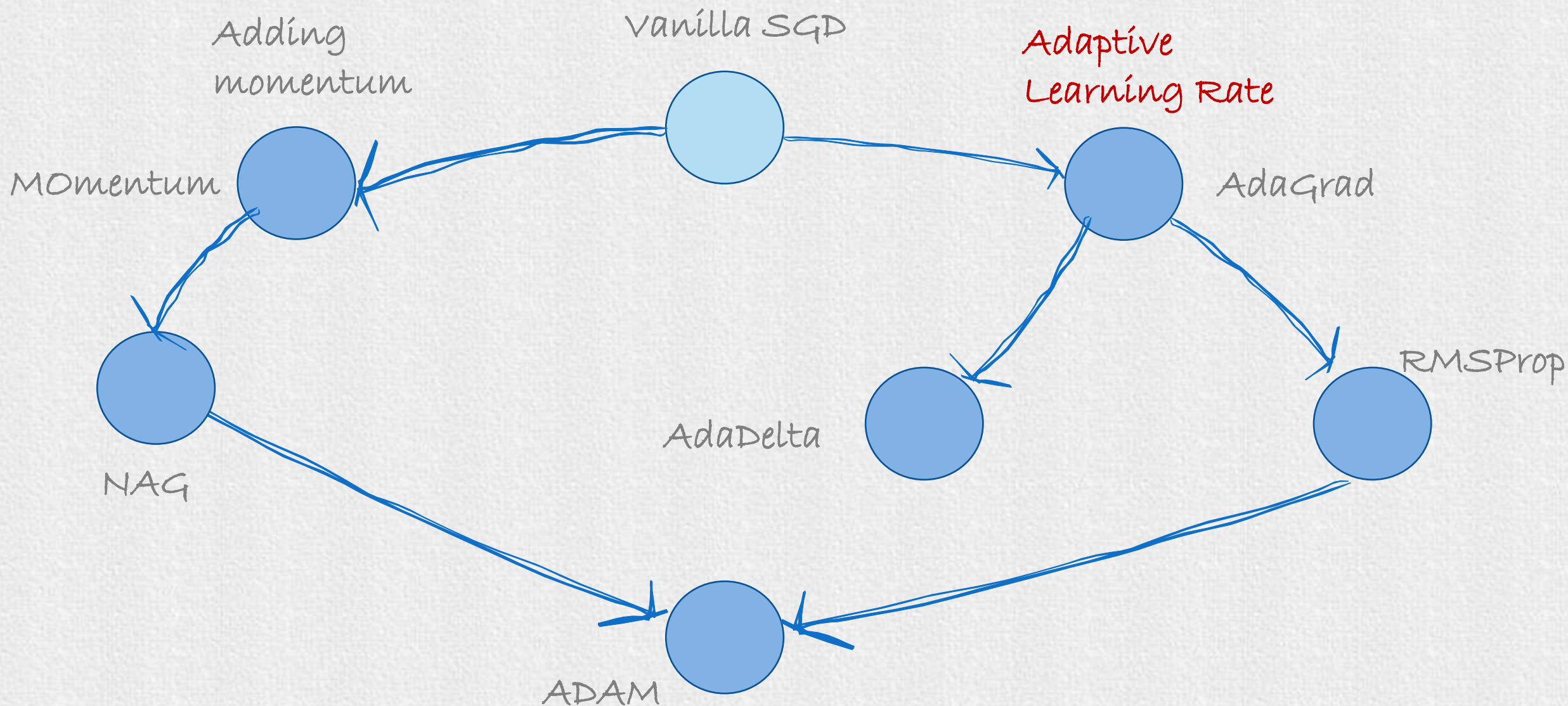


Nesterov momentum update



#17

## Diagram of Gradient Descent Development





#17

## Adaptive Gradient (Adagrad)

$$\theta_{t+1} = \theta - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_{\theta} J(\theta_t)$$

$$G_t = G_{t-1} + (\nabla_{\theta} J(\theta_t))^2$$

#17

## Adaptive Gradient (Adagrad)

$G$  keep increases, thus step size decays to zero !!

$$\theta_{t+1} = \theta - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_{\theta} J(\theta_t)$$

$$G_t = G_{t-1} + (\nabla_{\theta} J(\theta_t))^2$$



$$\theta_{t+1} = \theta - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_{\theta} J(\theta_t)$$

$$G_t = \gamma G_{t-1} + (1 - \gamma)(\nabla_{\theta} J(\theta_t))^2$$

#17

## AdaDelta

$$\theta_{t+1} = \theta_t - \Delta\theta$$

$$\Delta\theta = \frac{\sqrt{s + \epsilon}}{\sqrt{G + \epsilon}} \cdot \nabla_{\theta} J(\theta_t)$$

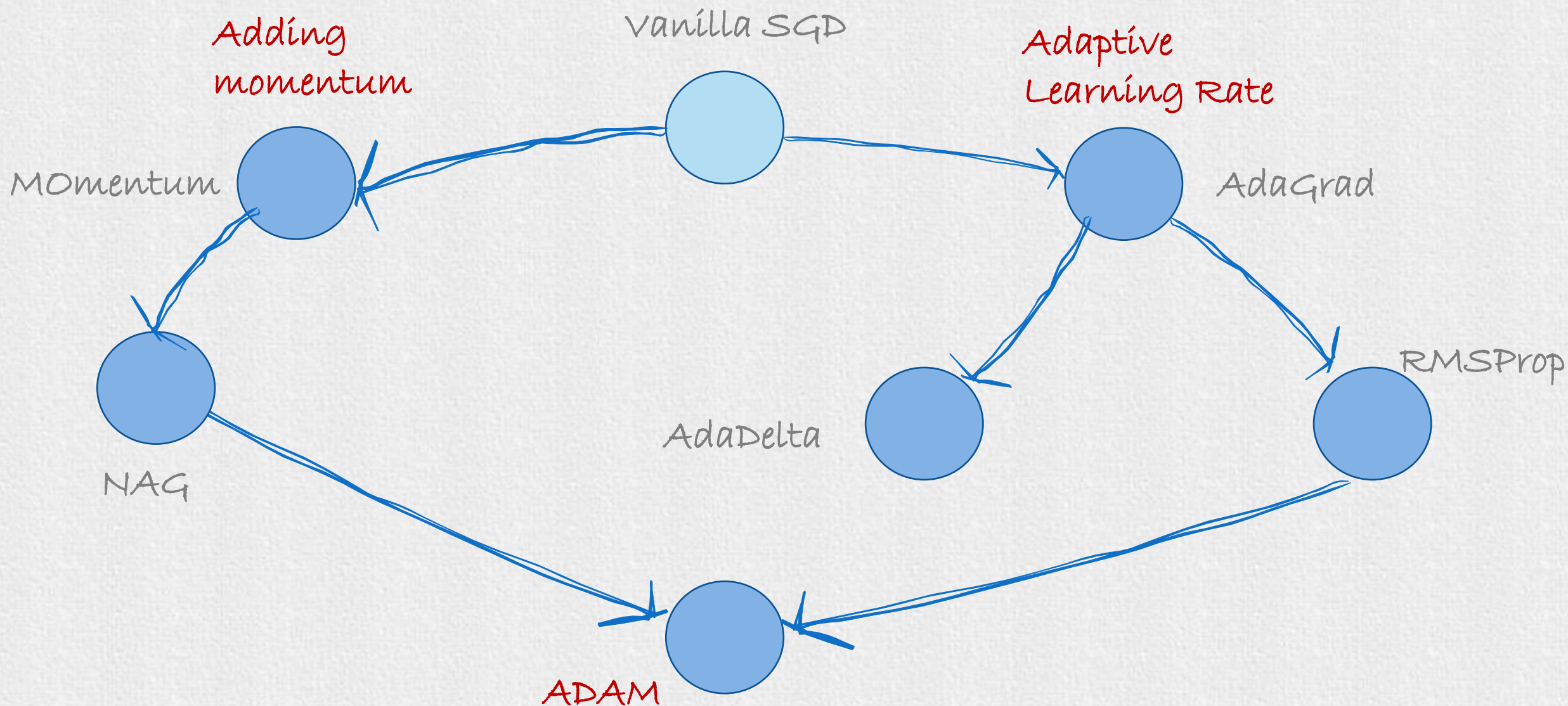
$$s_{t+1} = \gamma s_t + (1 - \gamma) \Delta\theta$$

$$G_{t+1} = \gamma G_t + (1 - \gamma) (\nabla_{\theta} J(\theta_t))^2$$



#17

## Diagram of Gradient Descent Development



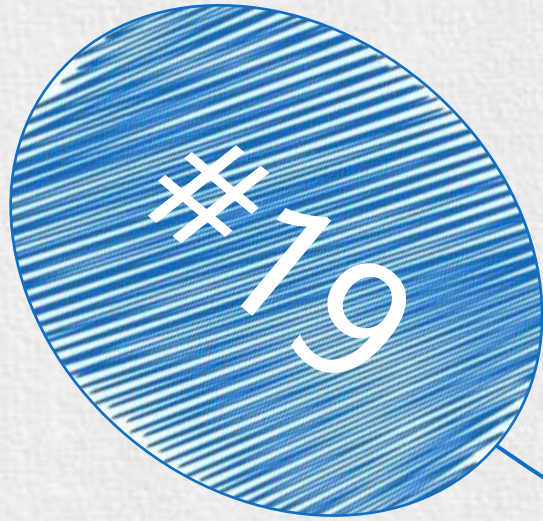
$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

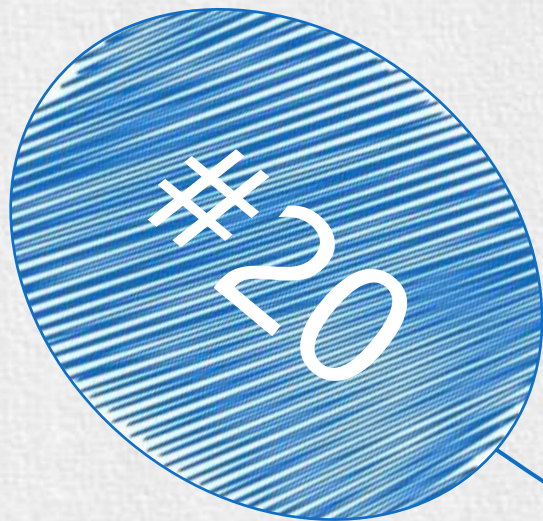




## *Assignment #3 Review*







## Topics to learn today

1. Review from last lecture
2. Problem of MLP
3. What is Convolutional Neural Network?
4. Implementing CNN with Pytorch







# *Problems of MLP*

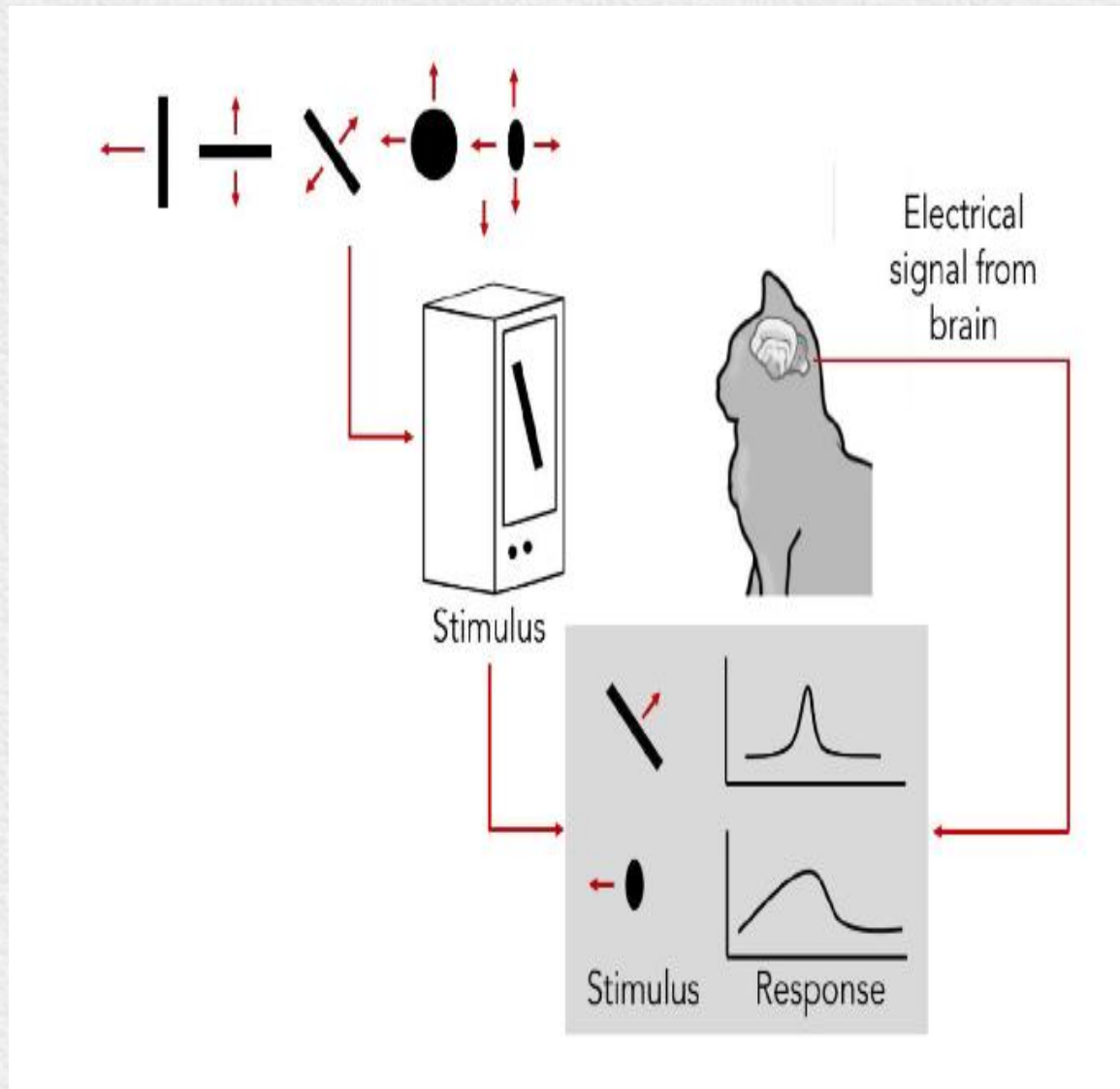
#20

## Number of parameters

- Since a neuron is connected with every neurons in preceding layer, number of parameters explodes as model gets deeper.
- Some of the parameters are meaningless.



# How human recognize an image?



## Hierarchical organization

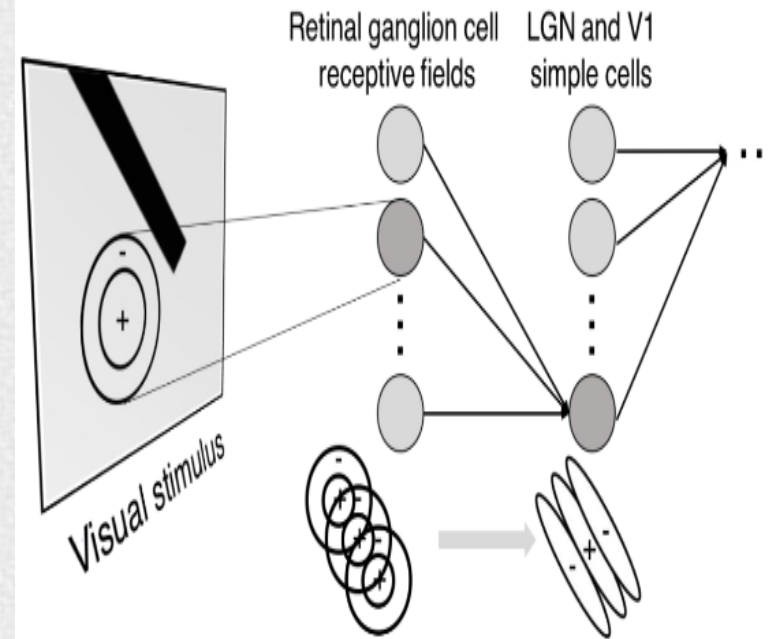
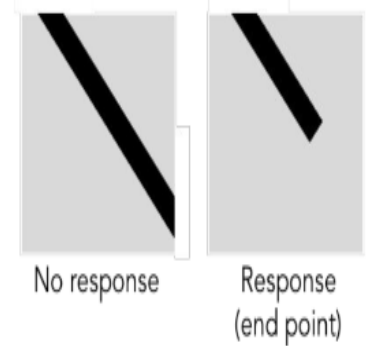


Illustration of hierarchical organization in early visual pathways by Lane McIntosh, copyright CS231n 2017

**Simple cells:**  
Response to light  
orientation

**Complex cells:**  
Response to light  
orientation and movement

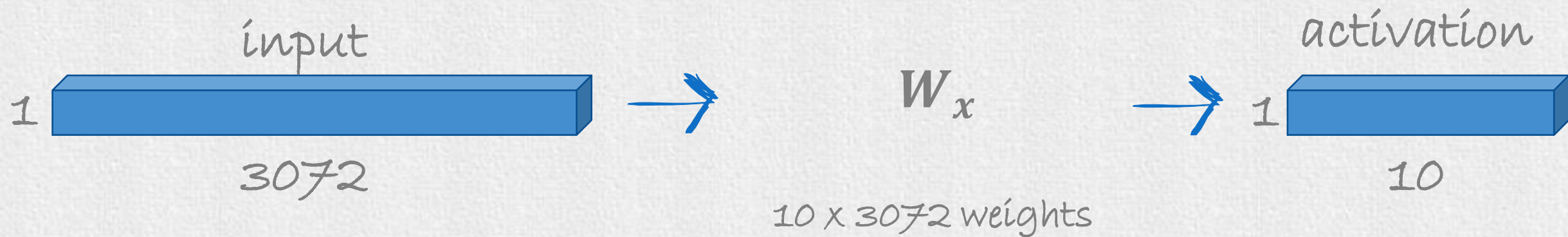
**Hypercomplex cells:**  
response to movement  
with an end point



#20

## MLP / Fully Connected Layer

$32 \times 32 \times 3$  image  $\rightarrow$  stretch to  $3072 \times 1$



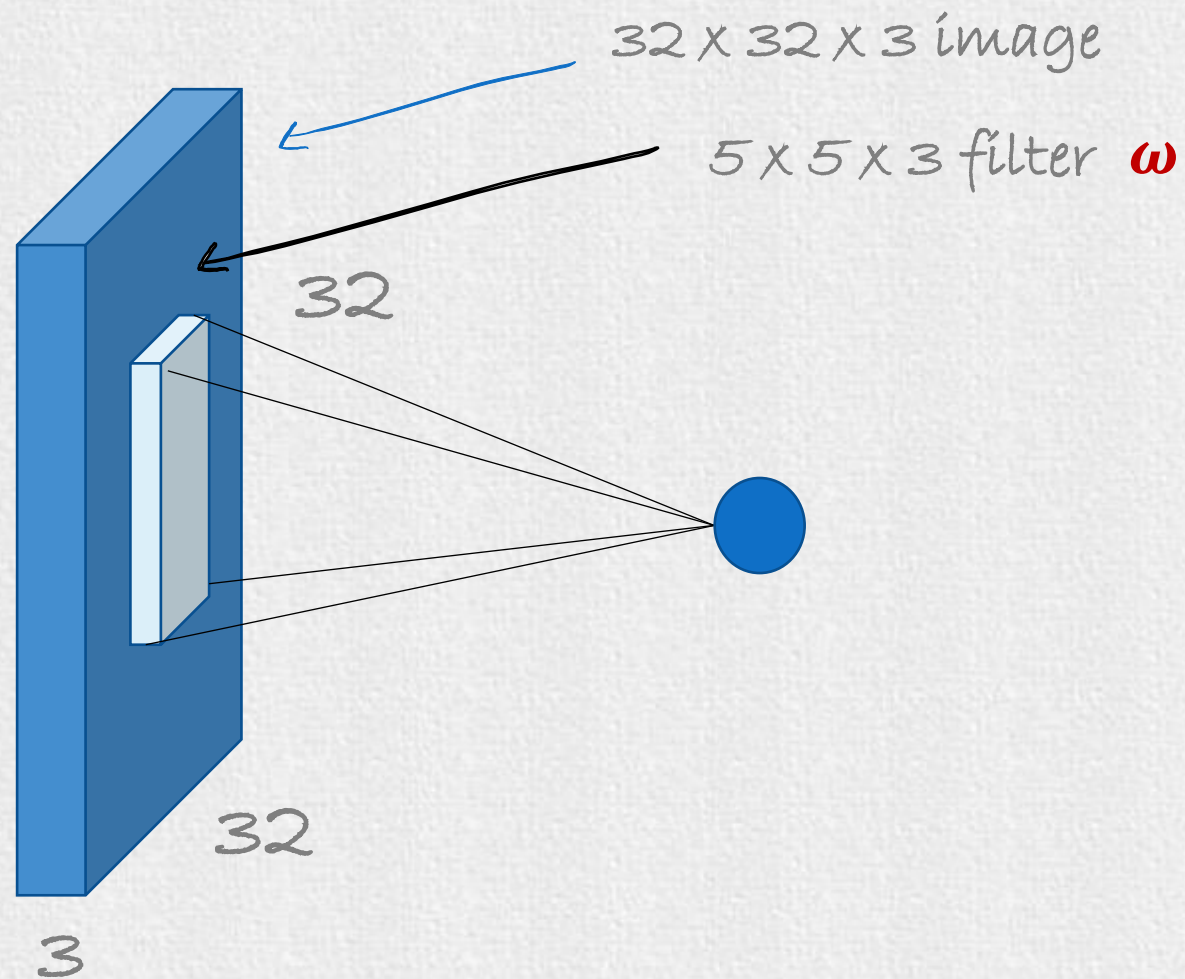




# What is Convolutional Neural Network?

#20

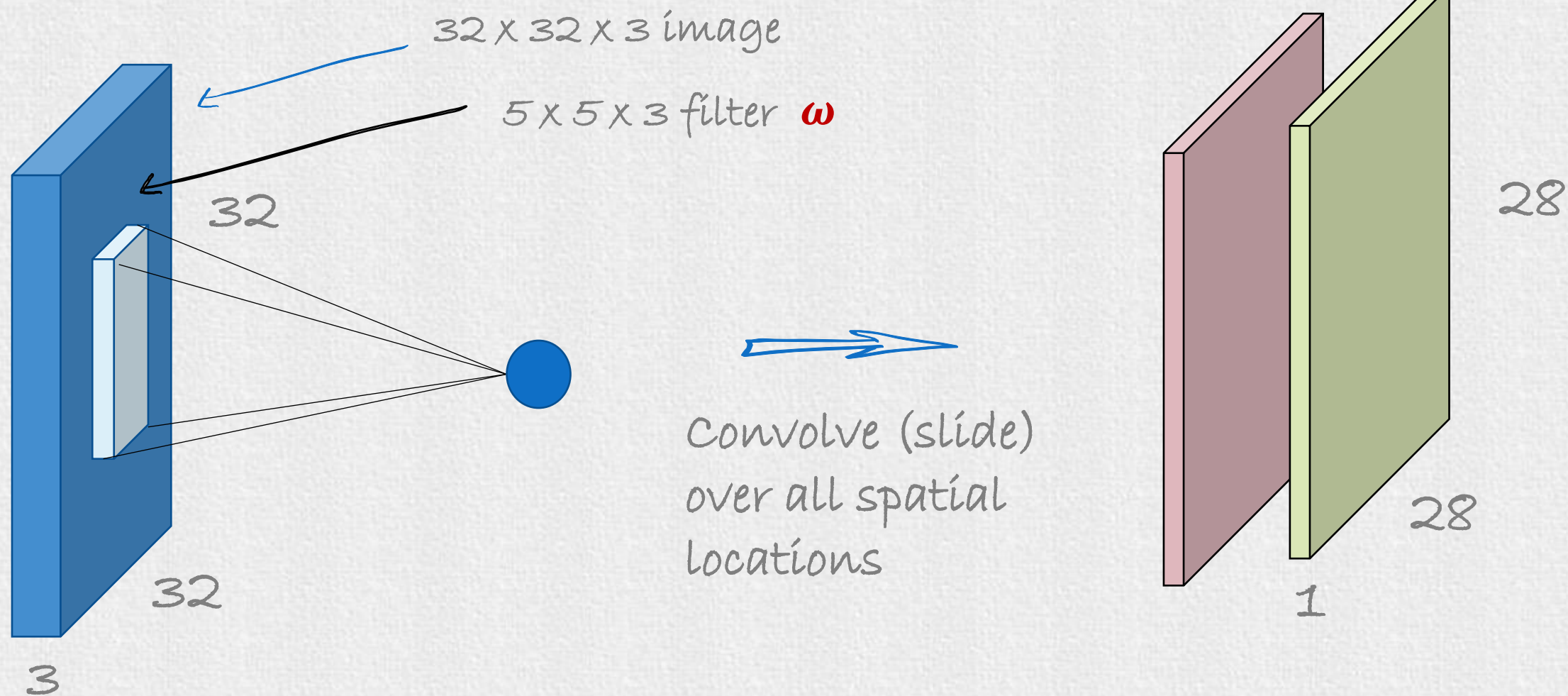
## Convolution Layer





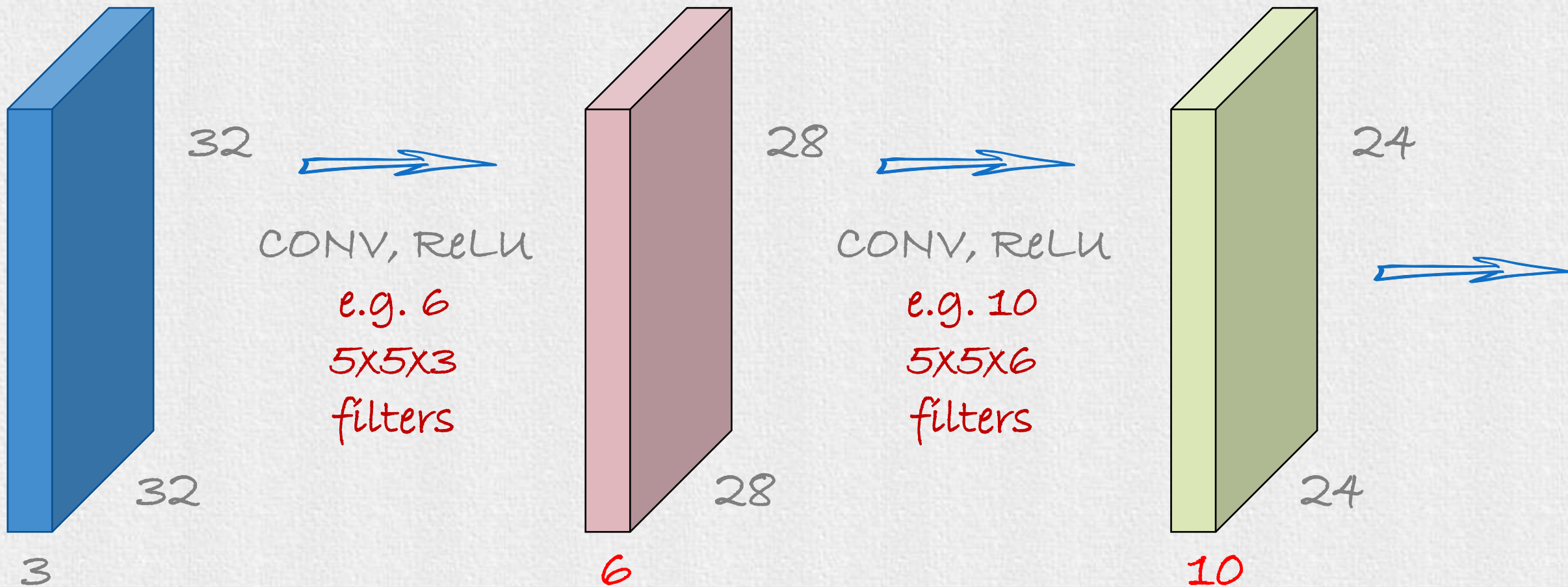
#20

## Convolution Layer



#20

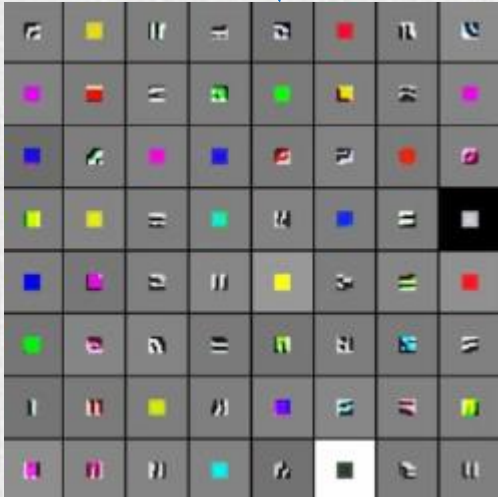
# Convolution Net



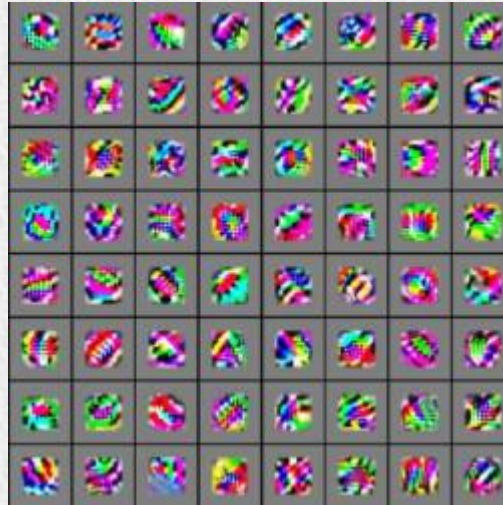




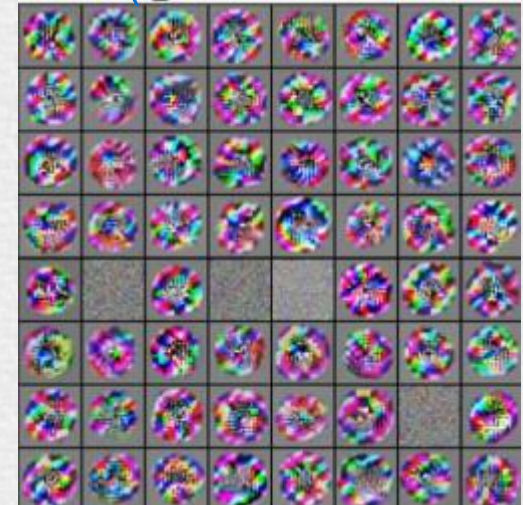
Low-level  
features



Mid-level  
features



High-level  
features



Linearly  
separable  
classifier

#20

## Calculating spatial dimension

7


7

7 x 7 input (spatially)  
assume 3 x 3 filter



#20

## Calculating spatial dimension

 $N$ 

			F			
	F					

 $N$ 

Output size:

$$(N - F) / \text{stride} + 1$$

e.g.  $N=7, F=3$ :

$$\text{Stride } 1 \Rightarrow (7 - 3) / 1 + 1 = 5$$

$$\text{Stride } 2 \Rightarrow (7 - 3) / 2 + 1 = 3$$

$$\text{Stride } 3 \Rightarrow (7 - 3) / 3 + 1 = 2.33$$

#20

## Zero padding

0	0	0	0	0	0			
0								
0								
0								
0								

e.g input  $7 \times 7$

$3 \times 3$  filter, applied with stride 1

Pad with 1 pixel border  $\Rightarrow$  what is output?

**$7 \times 7$  output!**

In general, common to see CONV layers with stride 1, filters of size  $F \times F$ , and zero-padding with  $(F-1)/2$ . (will preserve size spatially)

e.g.  $F = 3 \Rightarrow$  zero pad with 1

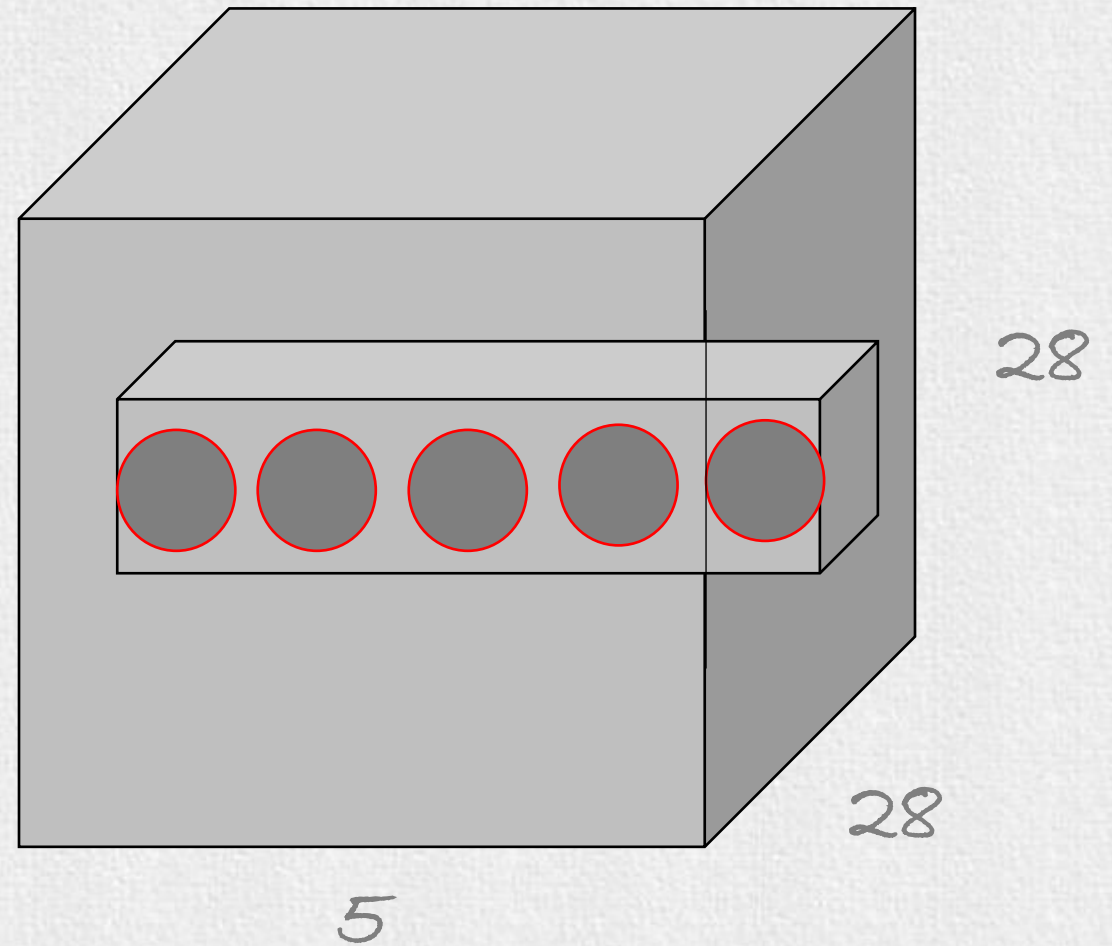
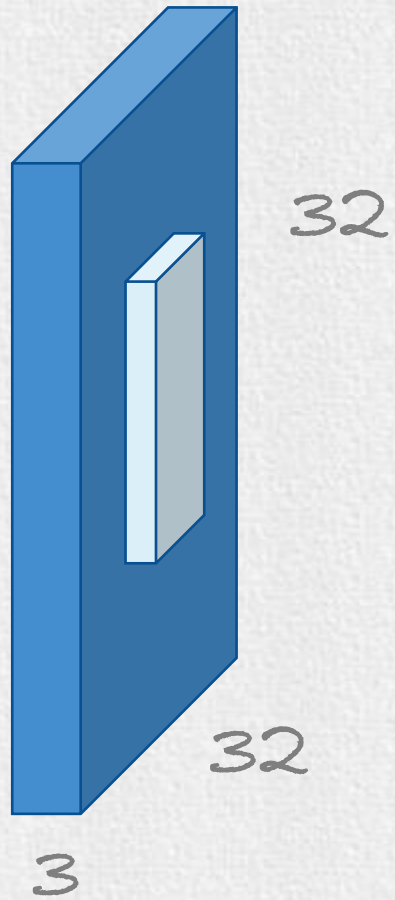
$F = 5 \Rightarrow$  zero pad with 2

$F = 7 \Rightarrow$  zero pad with 3



#20

## Neuron view of Convolutional Layer



The background of the slide is a large, irregular shape filled with dense, diagonal blue scribbles, resembling a hand-drawn or painted effect. The text is centered within this shape.

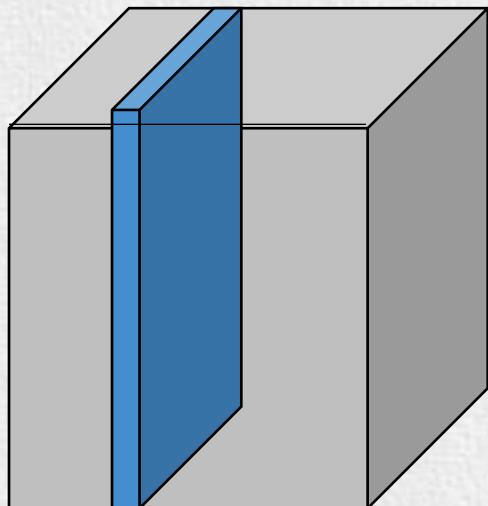
*Pooling and FC Layer*



#20

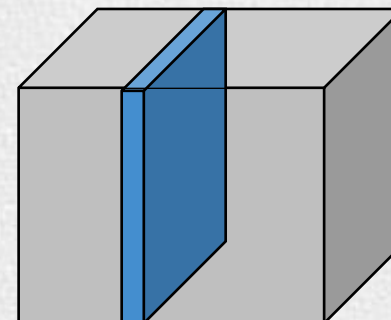
## Pooling layer

224x224x64

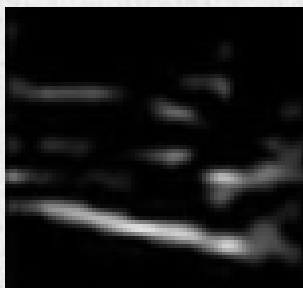


pool  
→

112x112x64

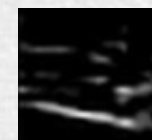


224



224

→  
downsampling



112

112

#20

## MAX POOLING

Single depth slice

x

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

y

Max pool with 2x2  
filters and stride 2

6	8
3	4



#20

Fully Connected Layer (FC)

MLP

#20

# Convolution Net

