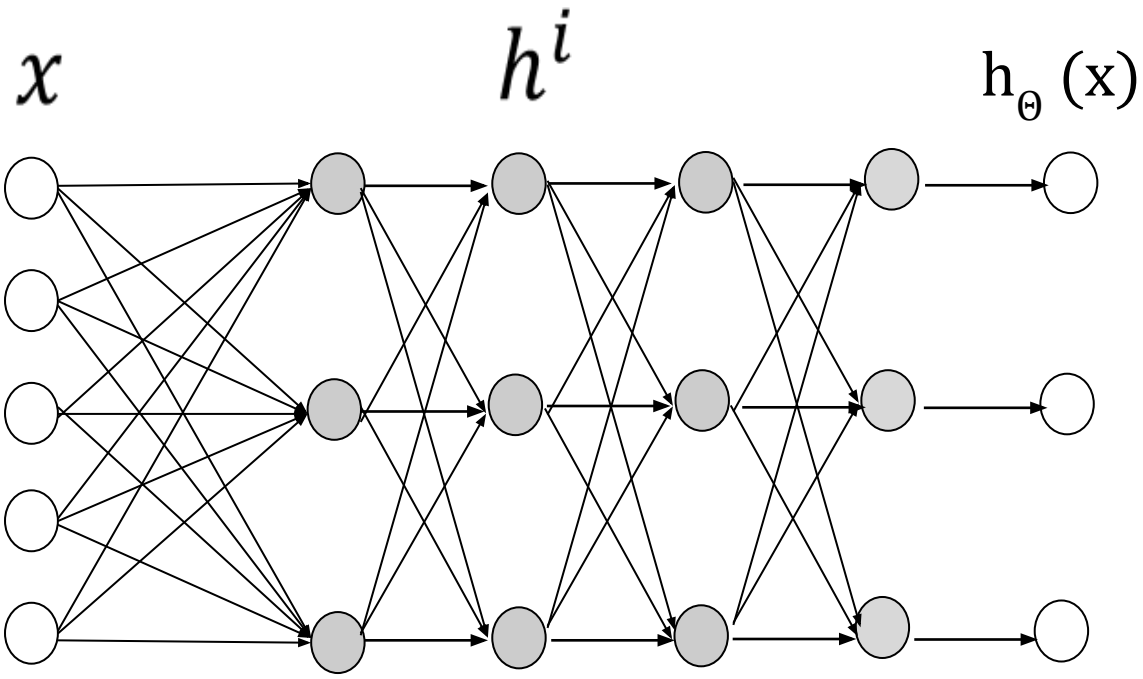
Announcements

- Pset-3 due 03/27
- Quiz-3 grades released tomorrow.

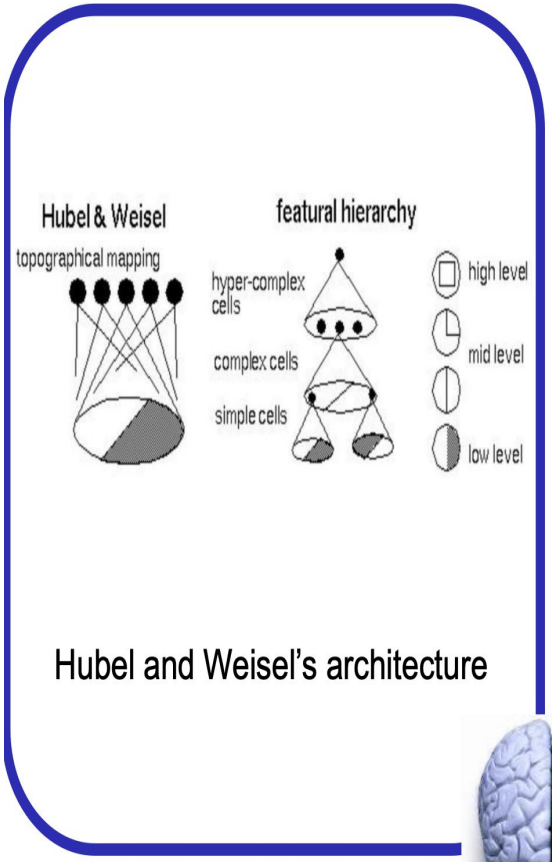
Quiz-3: Topics you want to be discussed in more detail

- Audio augmentation for video understanding
- Feature selection
- Clustering
- Image generation
- Backpropagation
- How to filter out the useless samples in the data set.
- More on Markov Chains
- Graphical models
- Applications using sound / vocal generation / learning spoken language
- Regression Analysis and its real word applications
- XGBoost, DenseNet, Confidence Intervals, U-Net
- Overview of the topics covered in each class.

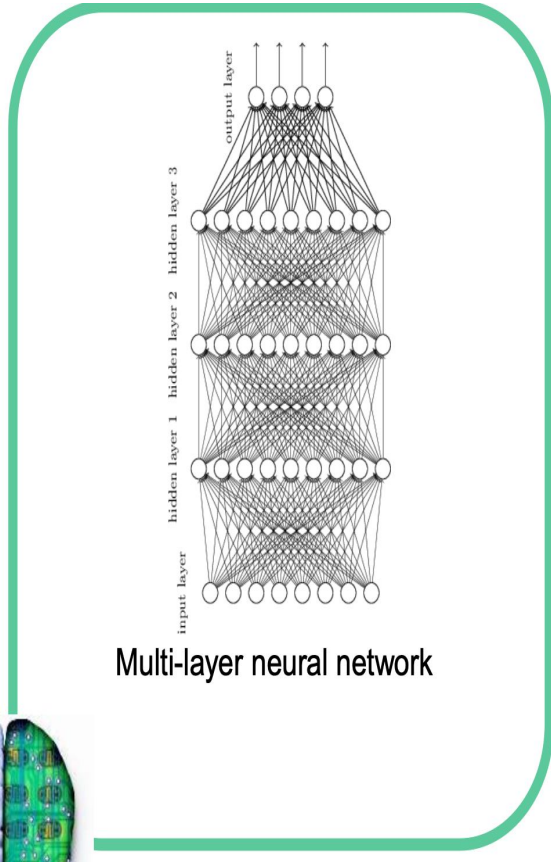
Last time: Neural networks



Question on hidden layers



Hubel and Weisel's architecture



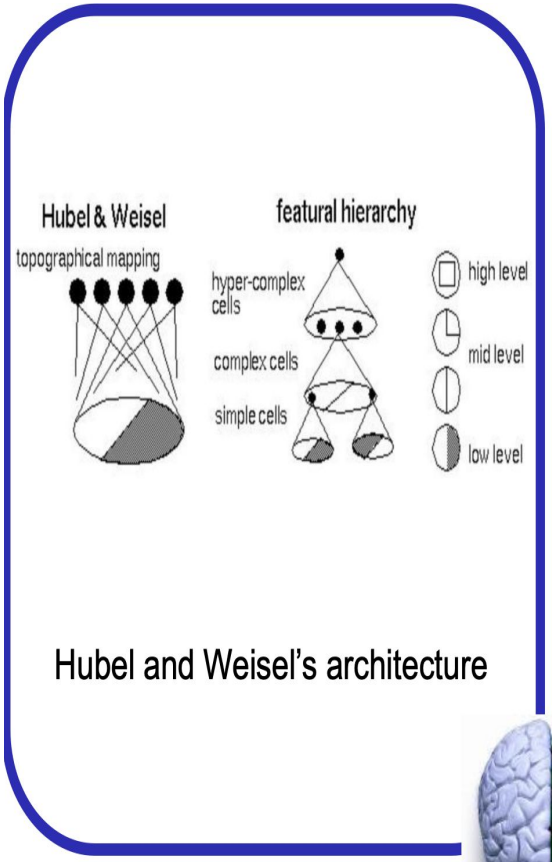
Multi-layer neural network

77

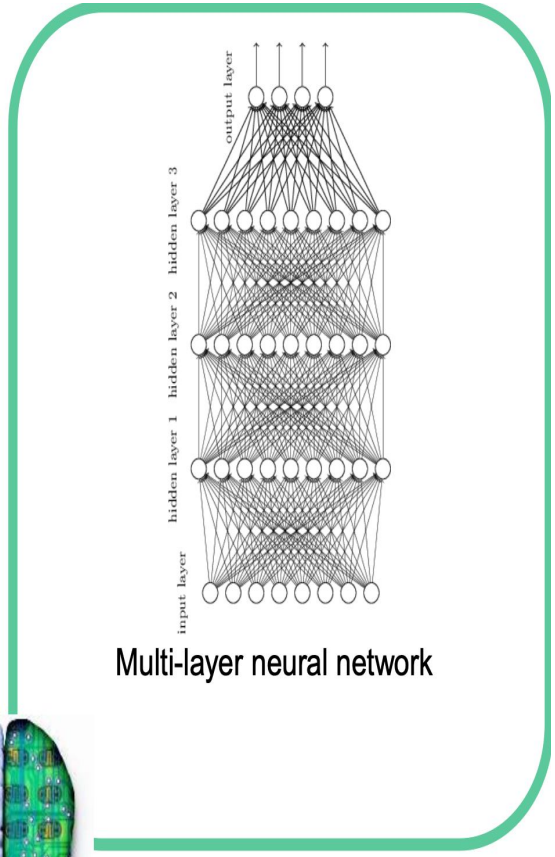
Adapted from Jia-bin Huang

- **Why are "hidden" layers called so?**
  - Just a nomenclature.
  - They do not directly interact with inputs or produce outputs.

Question on hidden layers



Hubel and Weisel's architecture



Multi-layer neural network

77

Adapted from Jia-bin Huang

- **Why are "hidden" layers called so?**
  - Just a nomenclature.
  - They do not directly interact with inputs or produce outputs.
- **Why are hidden layers compared to complex cells?**
  - Complex cells also receive signals processed from simple cells.
  - Complex cells also perform mid-level reasoning, like hidden layers.

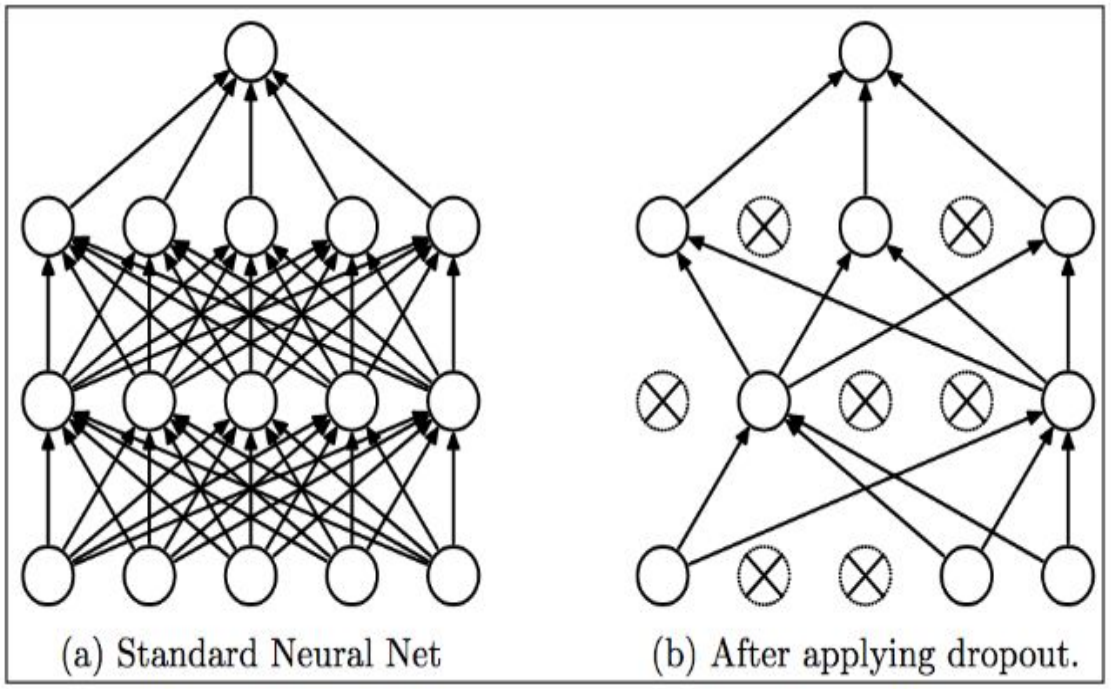
Last time

1. Network regularization
    a. Dropout
    b. Batch normalization
    c. Layer norm
    d. Group norm
2. Data Augmentation
    a. Image augmentations.
    b. Video augmentations.

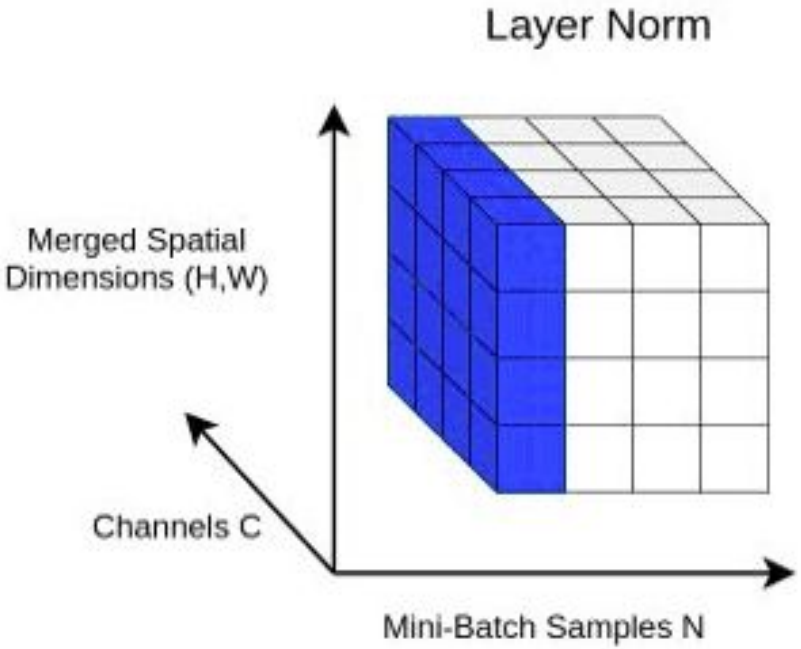
Regularizing Neural Nets (Dropout)

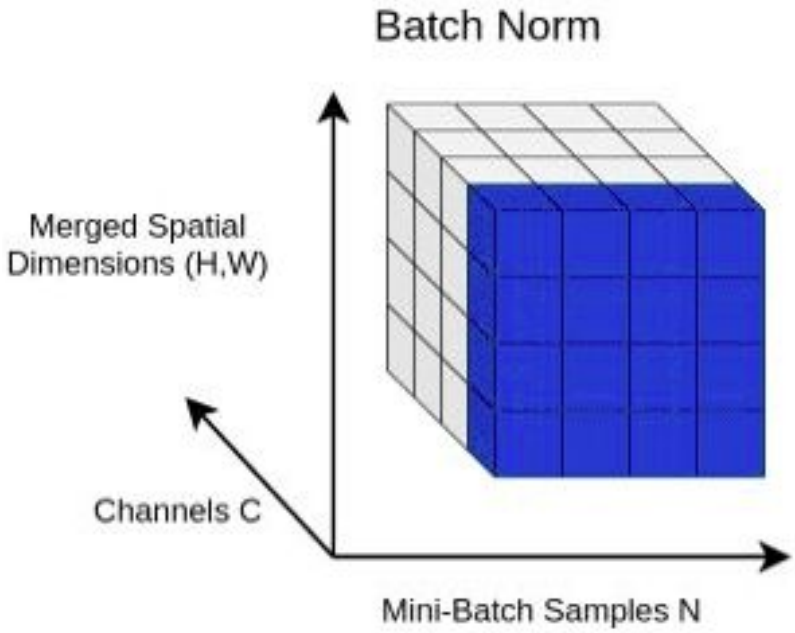
Issue: Some "neurons" might depend only on a handful of "neurons" from the last layer.

- We want diversity!
- Drop some connections during training.
- ***Use all connections at inference!***

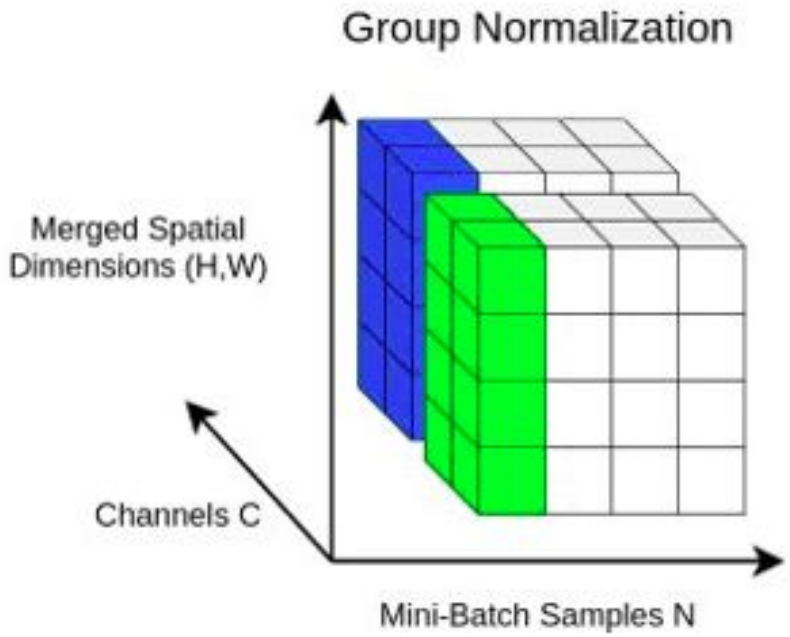


Different forms of normalizations

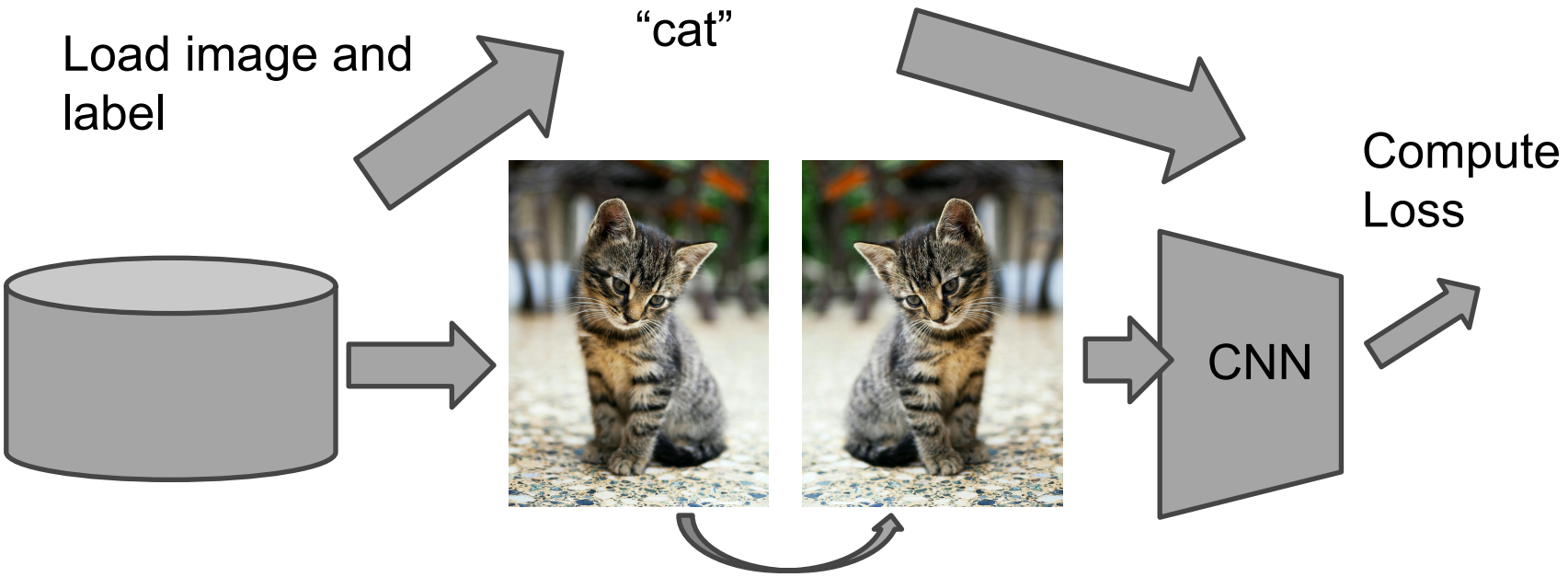




Different forms of normalizations



Data Augmentation



- Translation
- Rotation
- Color Jittering (randomize brightness, contrast, hue etc,)
- Stretching

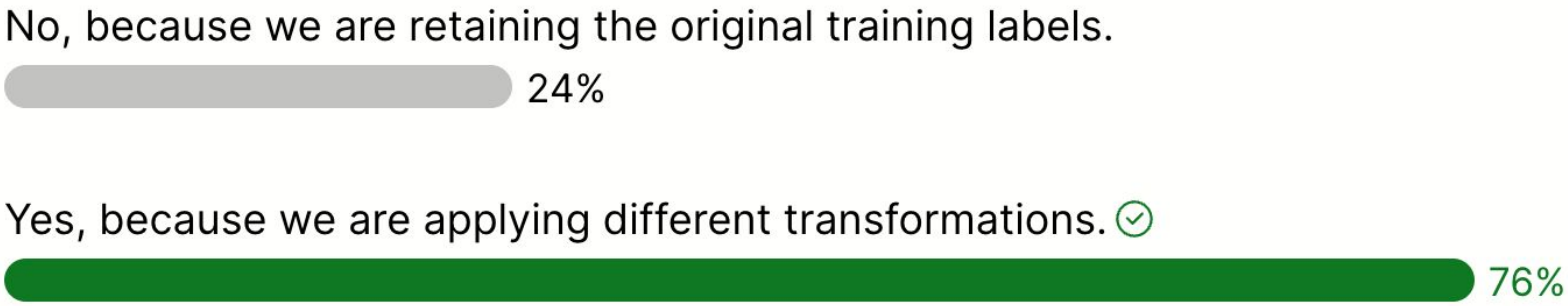
Types of video augmentations?

Last time

1. Network regularization
    a. Dropout
    b. Batch normalization
    c. Layer norm
    d. Group norm
2. Data Augmentation
    a. Image augmentations.
    b. Video augmentations.

Can image/text/audio data augmentation introduce noise in the training data?

Can image/text/audio data augmentation introduce noise in the training data?



Potential sources of noise

Label = Giraffe

Image size: 1080X720

Random cropping
(to size 224 X 224)

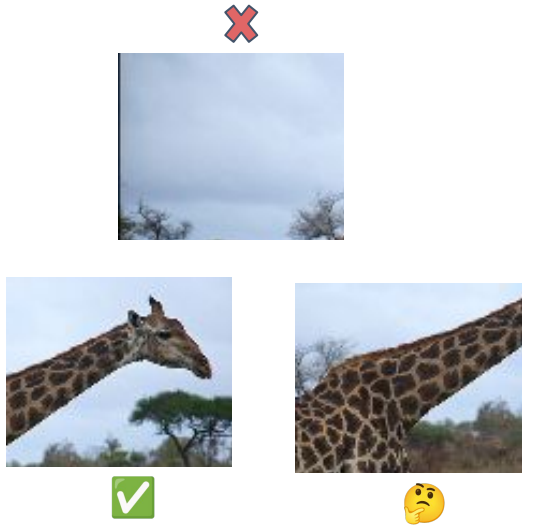
Potential sources of noise

Label = Giraffe

Image size: 1080X720

Random cropping
(to size 224 X 224)

Label = Giraffe



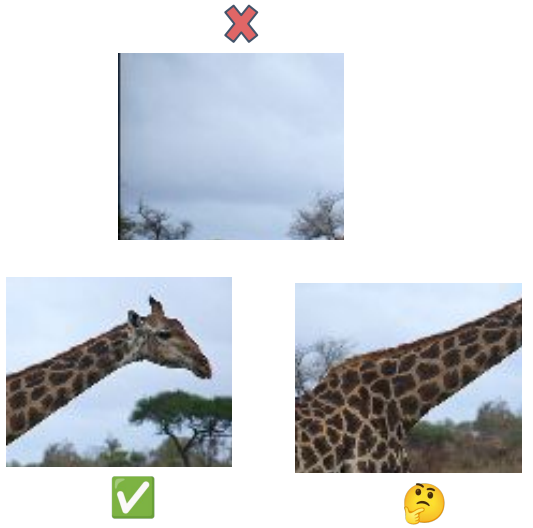
Potential sources of noise

Label = Giraffe

Image size: 1080X720

Random cropping
(to size 224 X 224)

Label = Giraffe



Default pipeline:

- First rescale the image to a fixed size (eg: 256 X 256)
- Then apply random crop of a fixed size (eg: 224 X 224)

Today

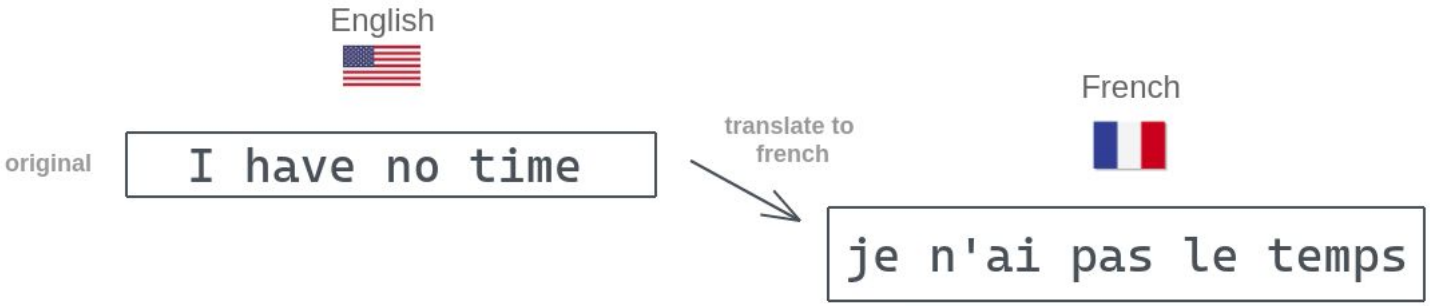
- Text augmentations
- Audio augmentations
- Convolutional networks

Today

- **Text augmentations**
- Audio augmentations
- Convolutional networks
  - Hidden layers
  - Receptive field
  - Strides

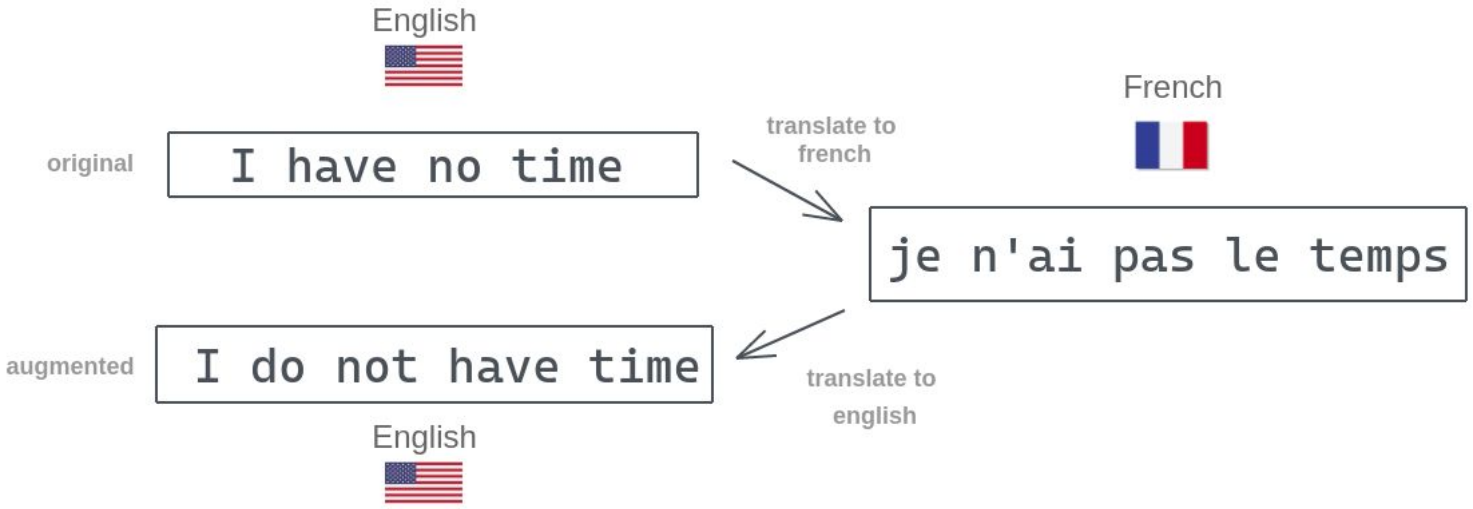
Text augmentations

- Reverse text translations



Text augmentations

- Reverse text translations



Text augmentations

- Synonym Replacement
  - *This **article** will focus on summarizing data augmentation **techniques** in NLP.*

  - *This **write-up** will focus on summarizing data augmentation **methods** in NLP.*

Text augmentations

- Inserting words that make sense in context.
  - *This **write-up** will focus on summarizing data augmentation **methods** in NLP.*

  - *This **write-up** will focus on summarizing data augmentation **methods** in NLP **methods***
- Removing words, swapping words, etc.

Today

- **Text augmentations**
  - Reverse text translations
  - Synonym replacement
  - Insert, remove, swap words
- Audio augmentations
- Convolutional networks
  - Hidden layers
  - Receptive field
  - Strides

Today

- Text augmentations
  - Reverse text translations
  - Synonym replacement
  - Insert, remove, swap words
- **Audio augmentations**
- Convolutional networks
  - Hidden layers
  - Receptive field
  - Strides

Audio data augmentations

1. **Noise injection**: add gaussian or random noise to the audio dataset to improve the model performance.
2. **Shifting**: shift audio left (fast forward) or right with random seconds.
3. **Changing the speed**: stretches times series by a fixed rate.
4. **Changing the pitch**: randomly change the pitch of the audio.
    a. perceived female vs male audio.
5. **Changing the accents** of a given audio signal

Audio data augmentations

1. **Noise injection**: add gaussian or random noise to the audio dataset to improve the model performance.
2. **Shifting**: shift audio left (fast forward) or right with random seconds.
3. **Changing the speed**: stretches times series by a fixed rate.
4. **Changing the pitch**: randomly change the pitch of the audio.
    a. perceived female vs male audio.
5. **Changing the accents** of a given audio signal

Why is this important?

Today

- Text augmentations
  - Reverse text translations
  - Synonym replacement
  - Insert, remove, swap words
- **Audio augmentations**
  - Noise injection
  - Change pitch, speed, accents
  - Change speed
- Convolutional networks
  - Hidden layers
  - Receptive field
  - Strides

Summary: Data augmentations

Image and video
Spatial translation, rotation
Stretching
Add noise: gaussian, salt and pepper
Randomize brightness, contrast, hue etc.
Slow down/ speed up video

Summary: Data augmentations

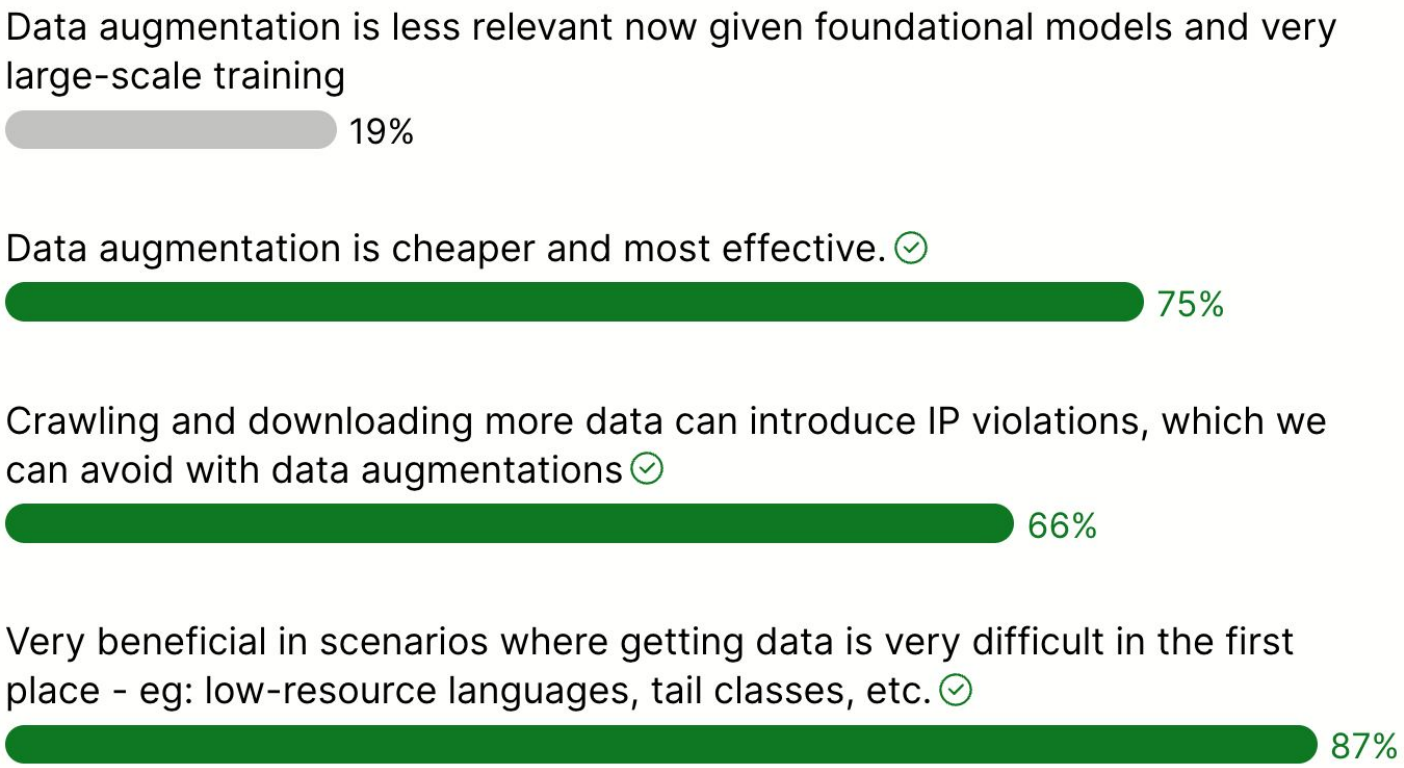
Image and video	Text
Spatial translation, rotation	Reverse text translations
Stretching	
Add noise: gaussian, salt and pepper	Insert, remove, swap words
Randomize brightness, contrast, hue etc.	Synonym replacement
Slow down/ speed up video	

Summary: Data augmentations

Image and video	Text	Audio
Spatial translation, rotation	Reverse text translations	
Stretching		Shifting the audio signal
Add noise: gaussian, salt and pepper	Insert, remove, swap words	Noise injection
Randomize brightness, contrast, hue etc.	Synonym replacement	Changing the pitch, accents
Slow down/ speed up video		Changing audio speed

Which of the following is/are true about data augmentation?

Which of the following is/are true about data augmentation?



Today

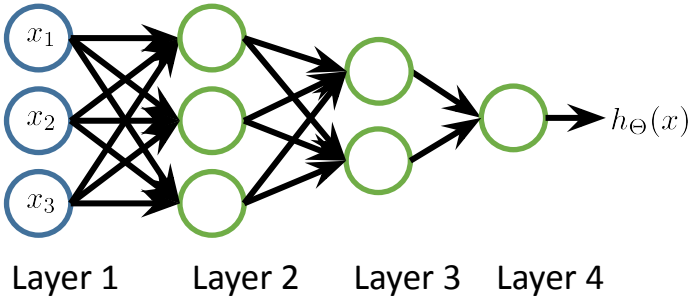
1. Network regularization
   a. Dropout
   b. Batch normalization
   c. Layer norm
   d. Group norm
2. Data Augmentation
   a. Image, video, text, audio augmentations.
3. **Convolutional Neural Networks**

Architecture agnostic

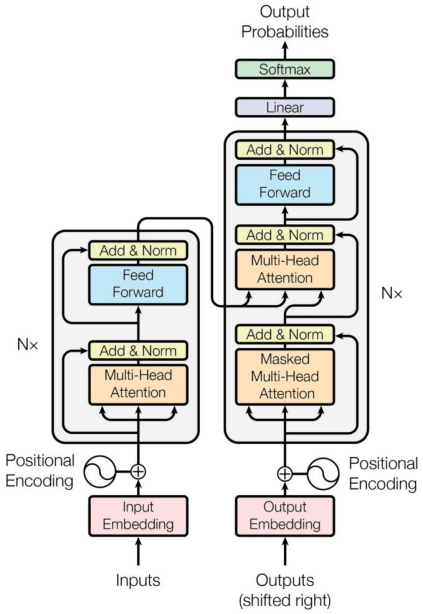
Network architectures

Feed-forward

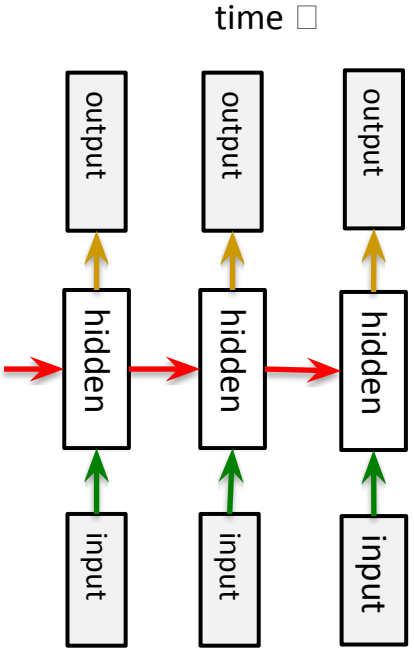
Fully connected



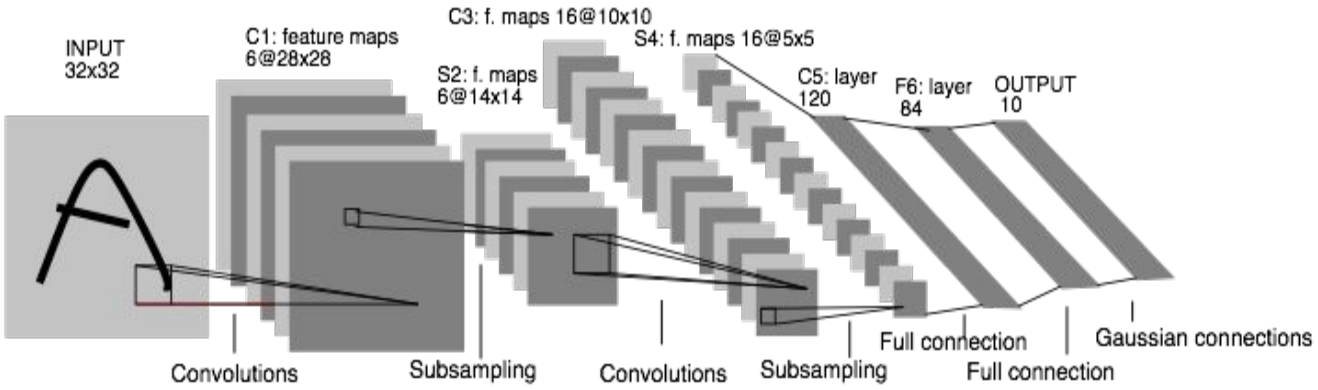
Transformer



Recurrent

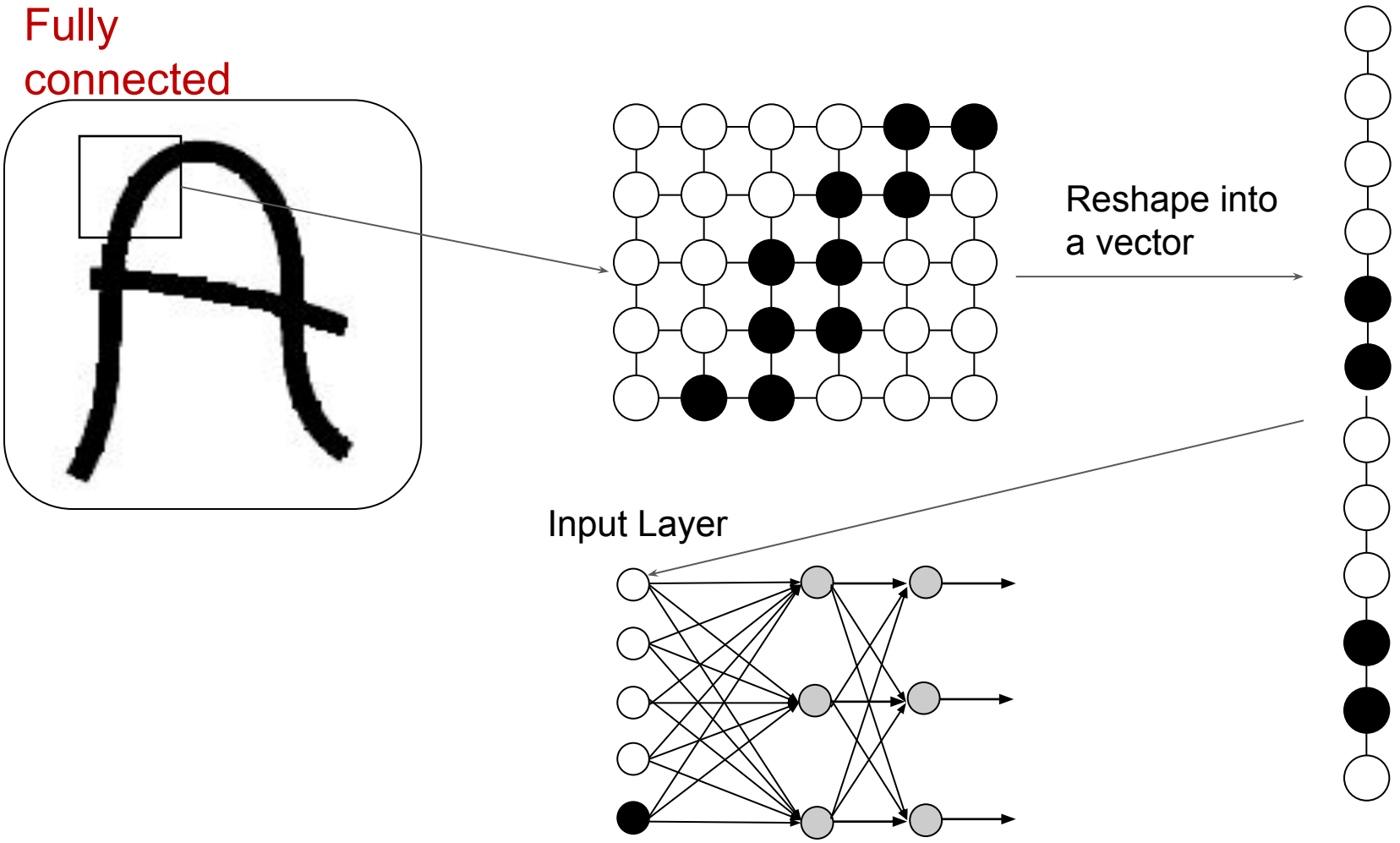


Convolutional



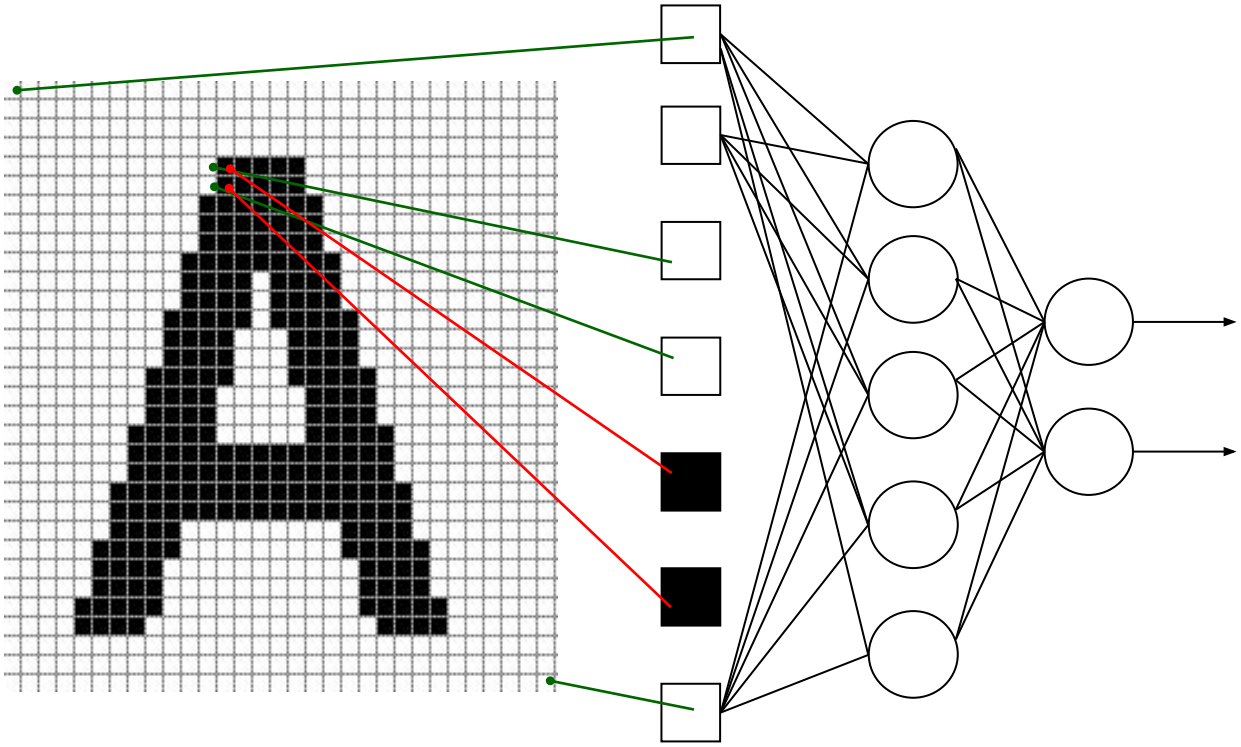
Representing images

Fully connected



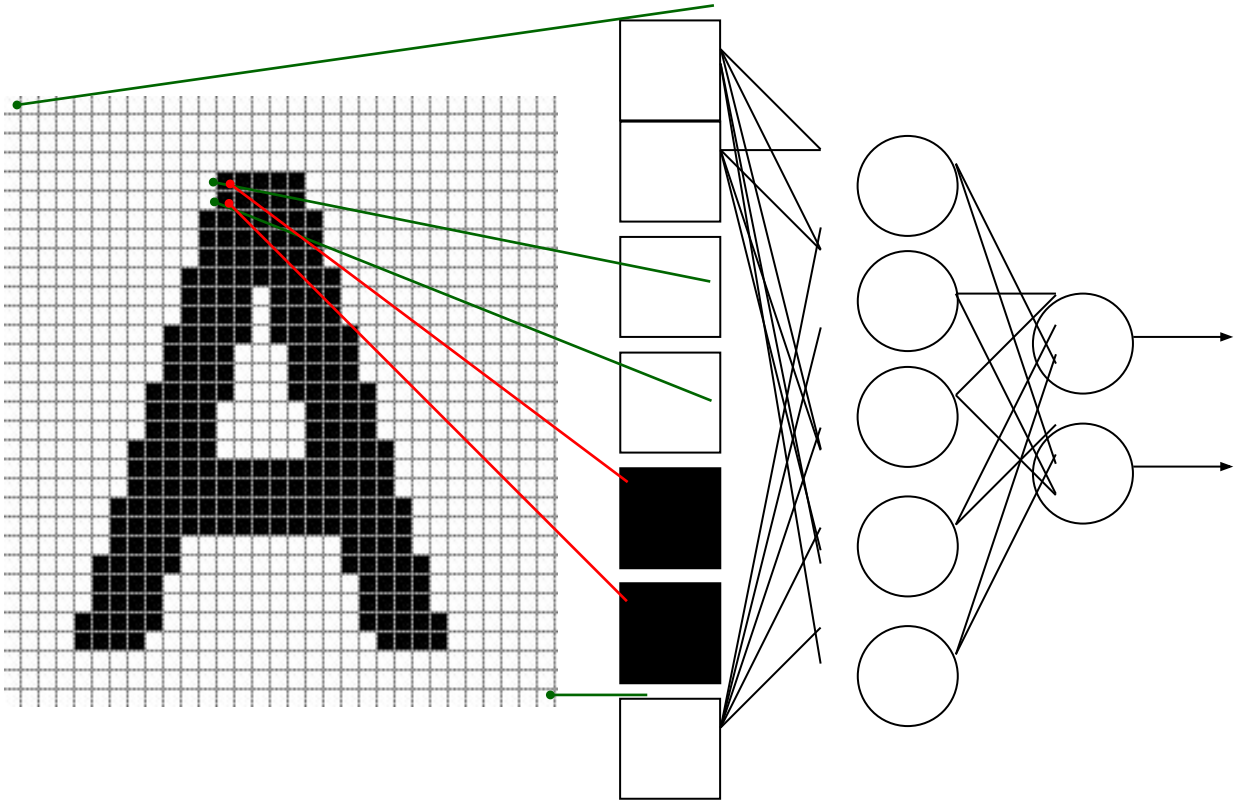
2D Input: fully connected network

Vectorize input by copying rows into a single column



2D Input: fully connected network

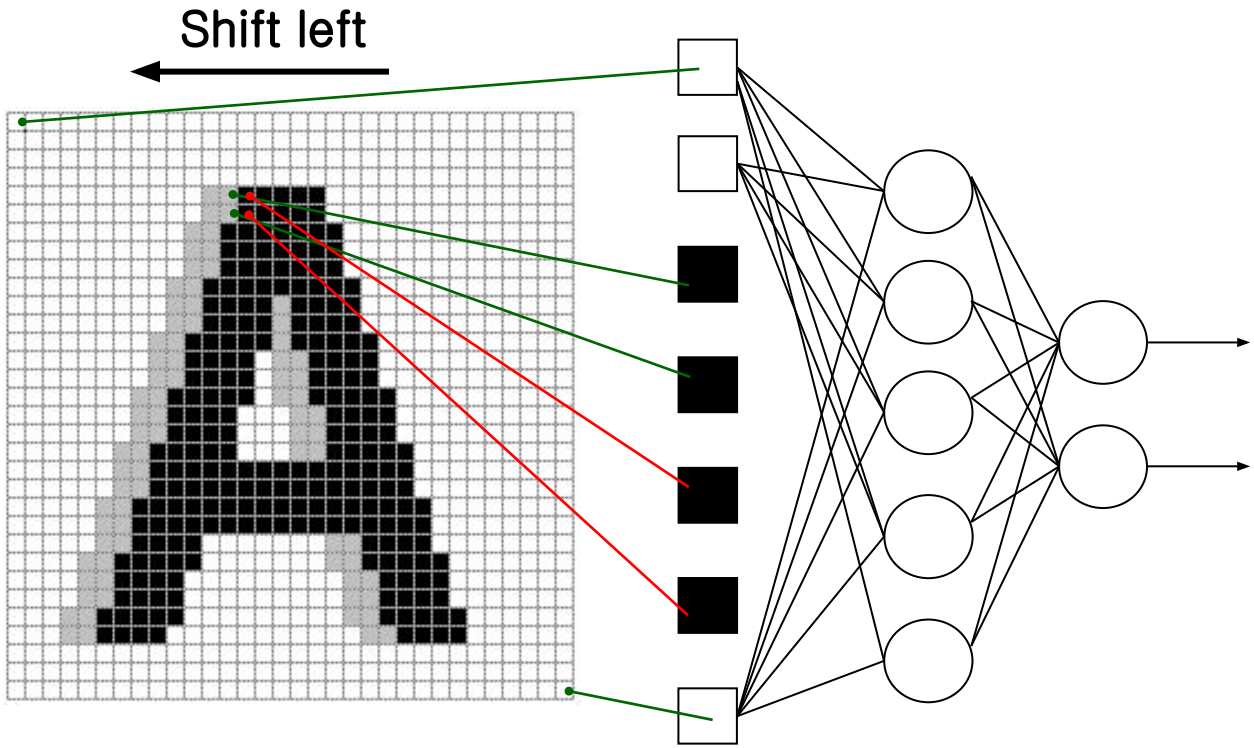
Vectorize input by copying rows into a single column



Can this lead to any issues?

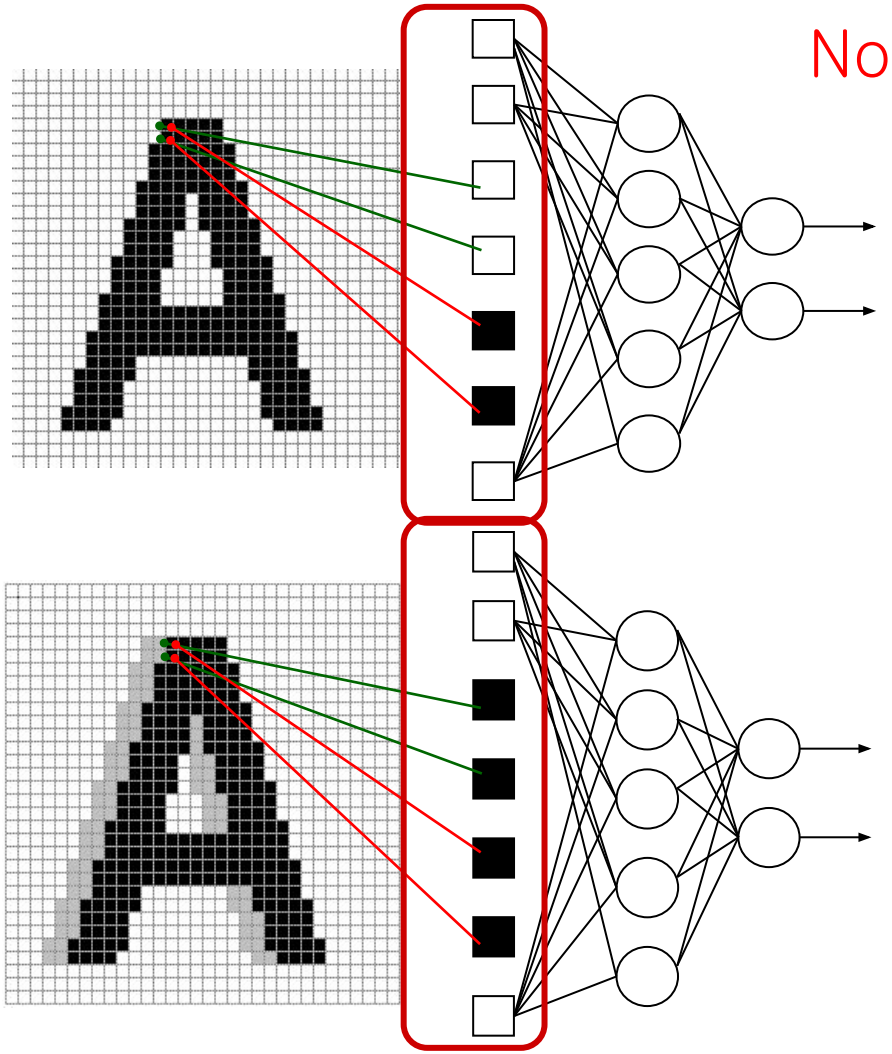
2D Input: fully connected network

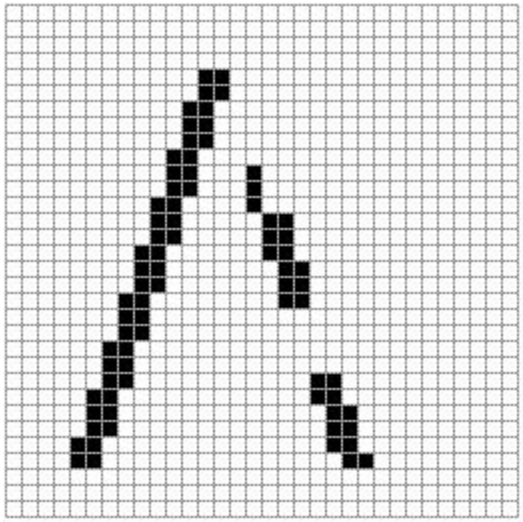
Problem: shifting, scaling, and other distortion changes location of features



2D Input: fully connected network

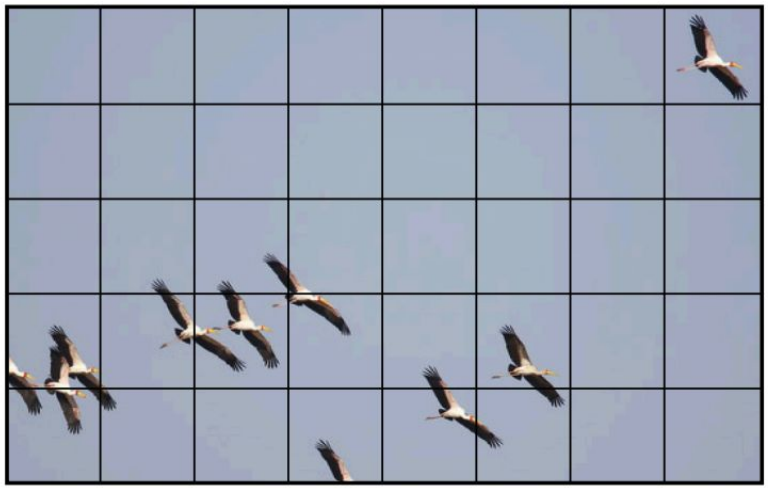
Not invariant to translation!





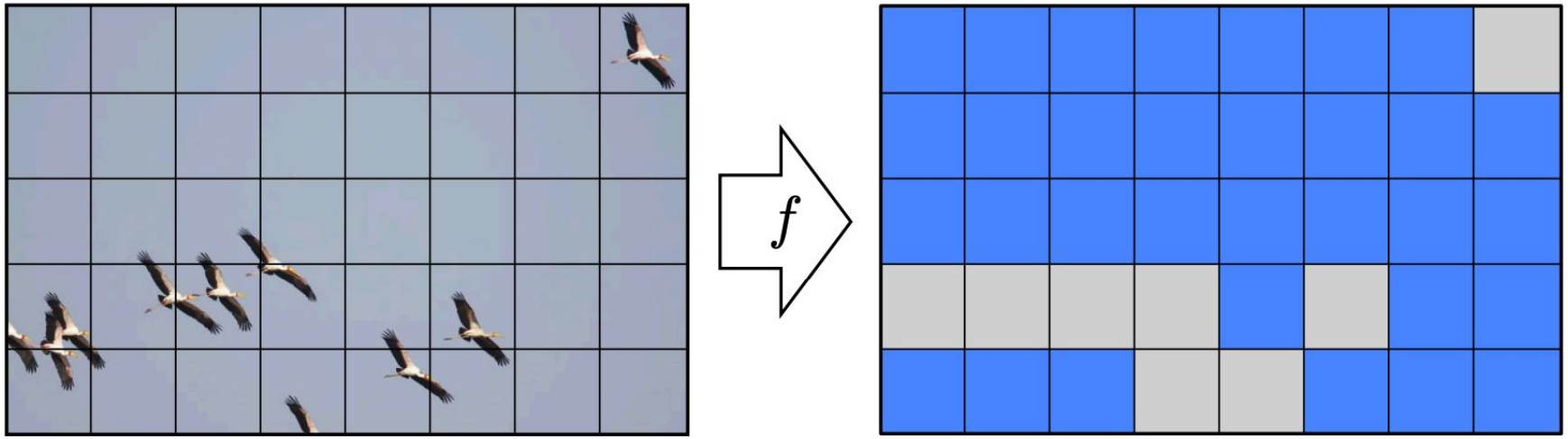
154 inputs changed just from 2 pixels shifting left
77 : black to white
77 : white to black

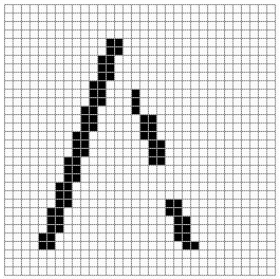
~~Pixels~~ cell-based representation



Sky	Sky	Sky	Sky	Sky	Sky	Sky	Bird
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Bird	Bird	Bird	Sky	Bird	Sky	Sky	Sky
Sky	Sky	Sky	Bird	Sky	Sky	Sky	Sky

~~Pixels~~ cell-based representation



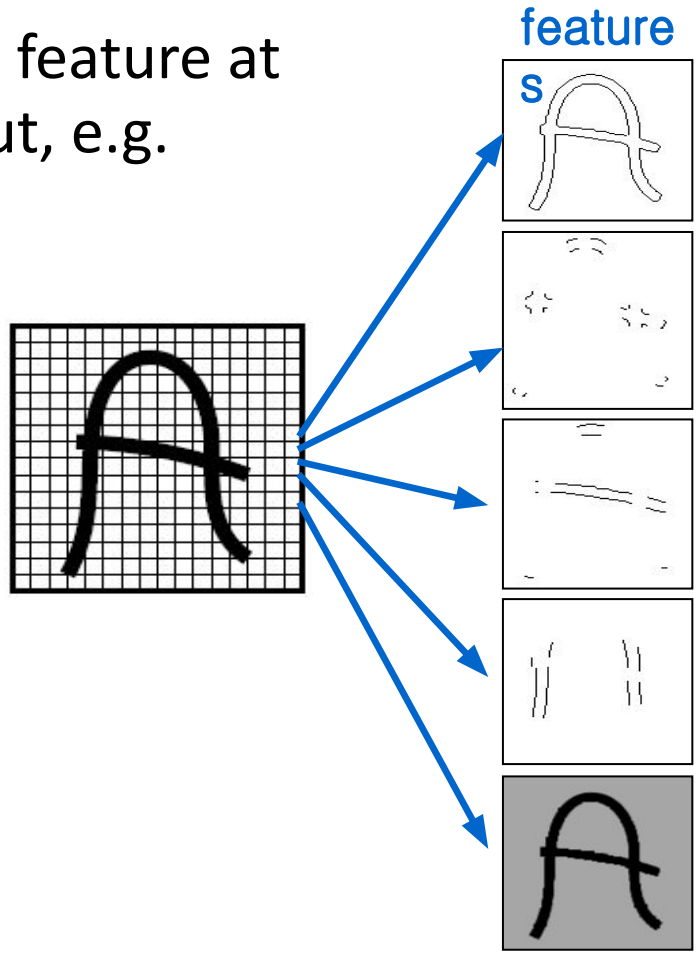


Potentially better than pixel based representation.

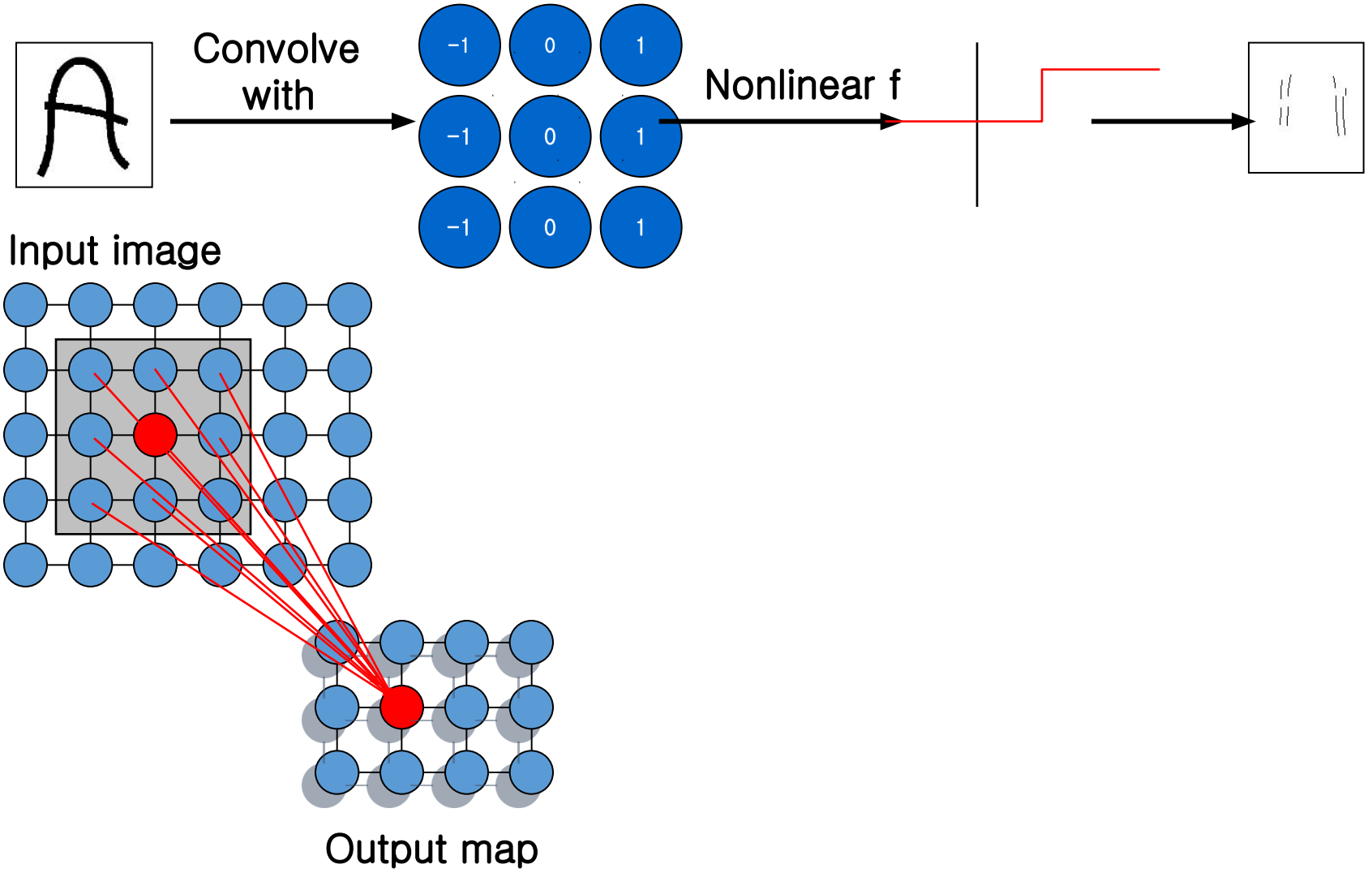
Problem: shifting, scaling, and other distortion changes location of features

Solution: **Convolution layer**

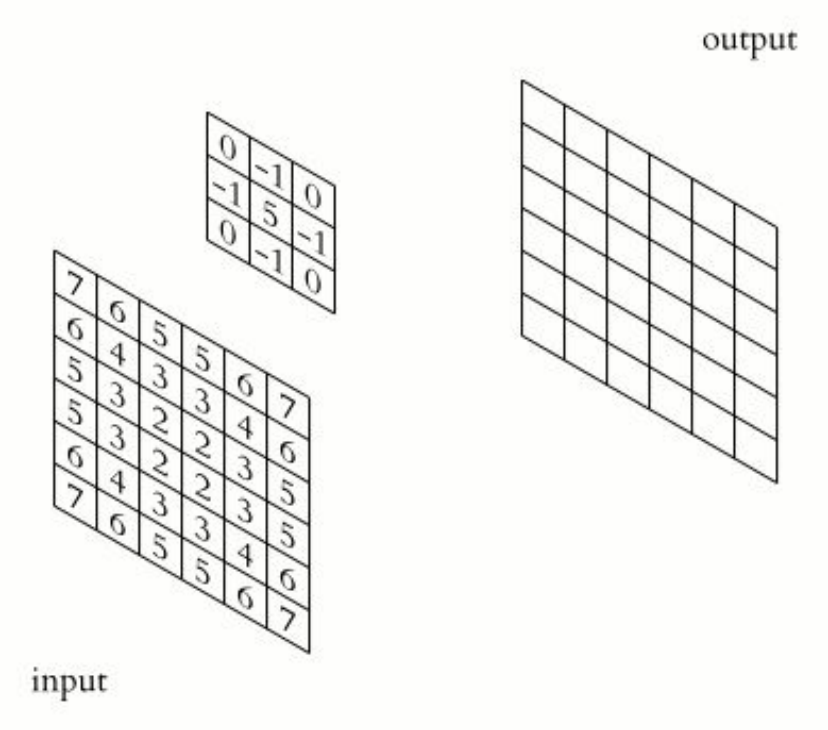
- **Motivation:** Detect the same feature at different positions in the input, e.g. image
- Preserve input topology



Convolution layer in 2D

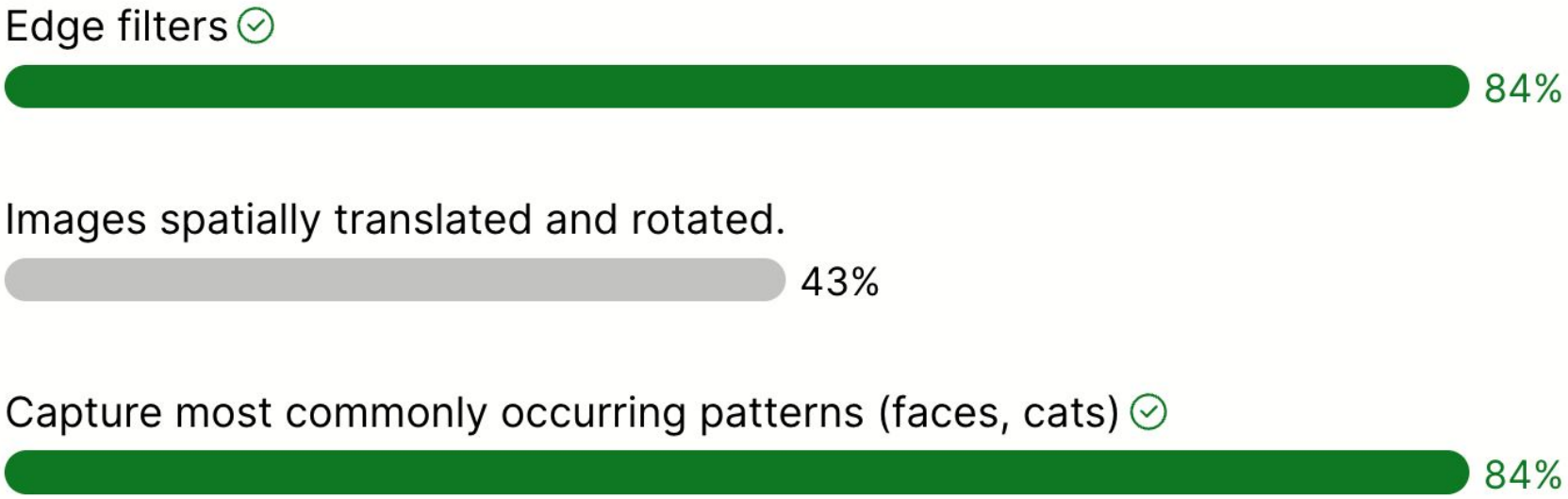


Convolution layer in 2D



What do output feature maps look like? Select all that apply

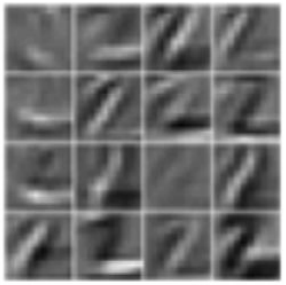
What do output feature maps look like? Select all that apply



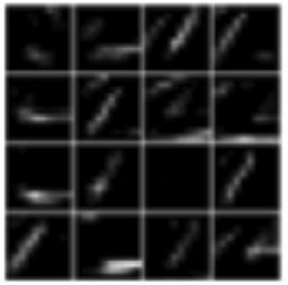
Feature maps

conv1

relu1

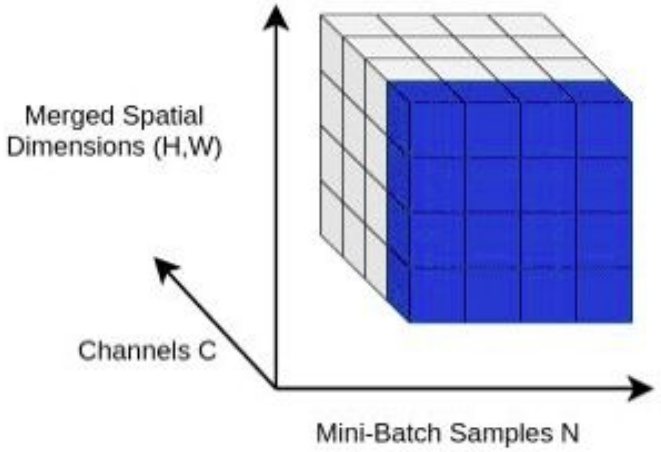


conv2

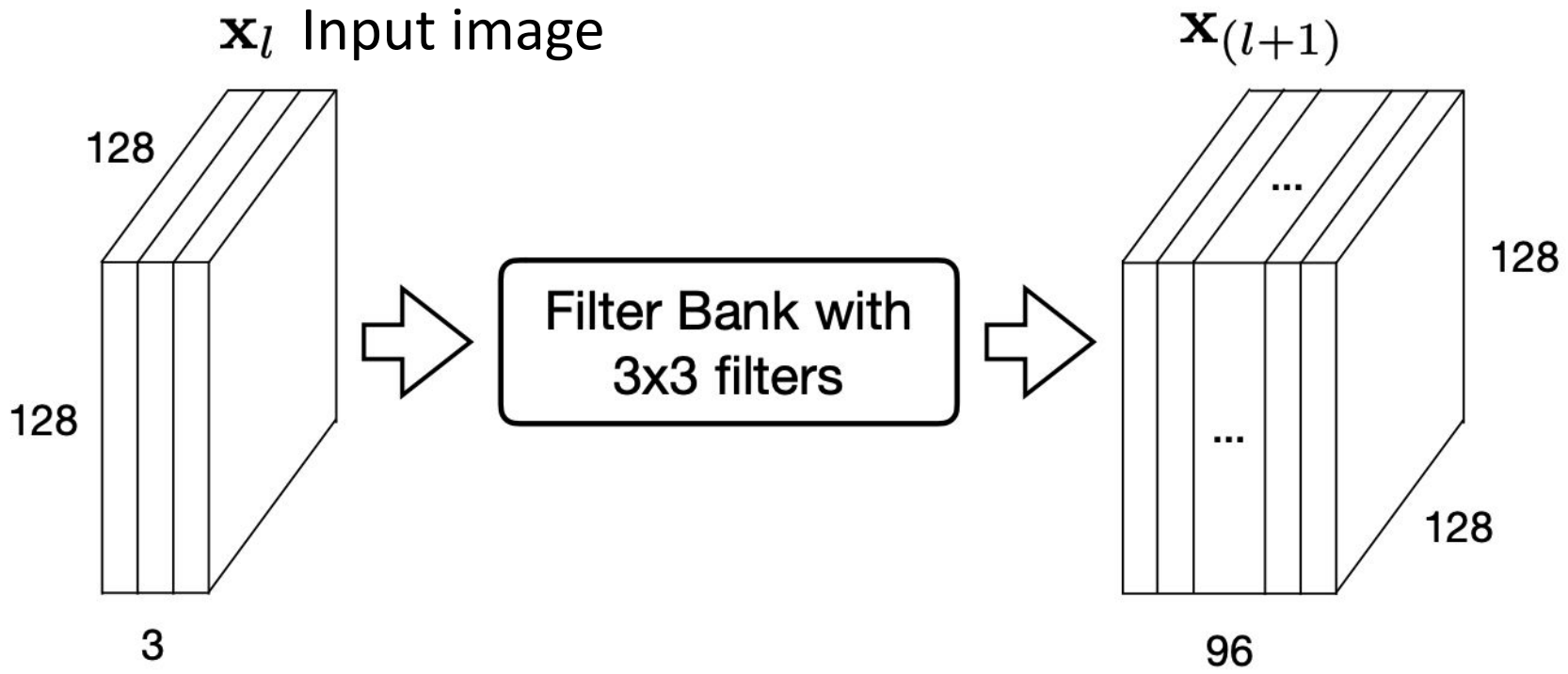


relu2

- Each layer can be thought of as a set of C **feature maps** aka **channels**
- Each feature map is an NxM image



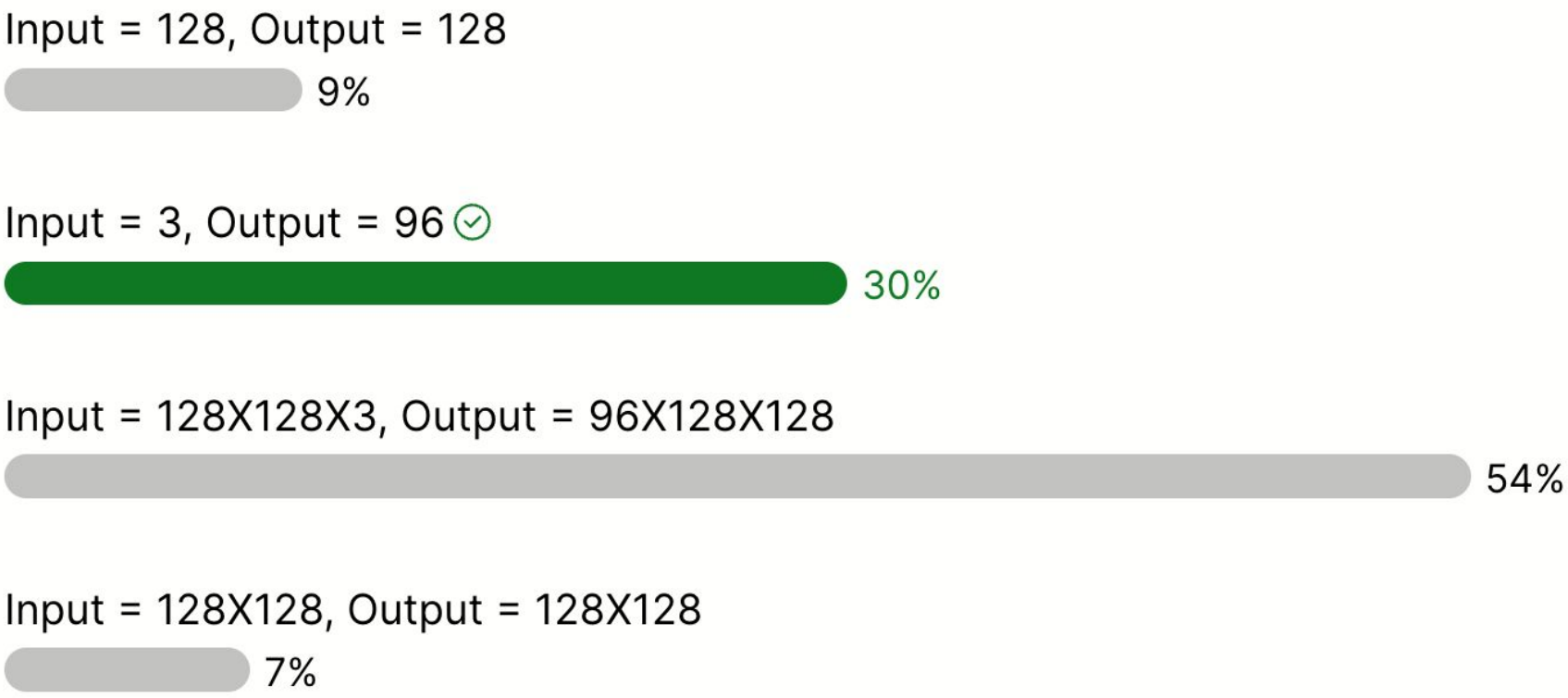
Multiple channels: Example



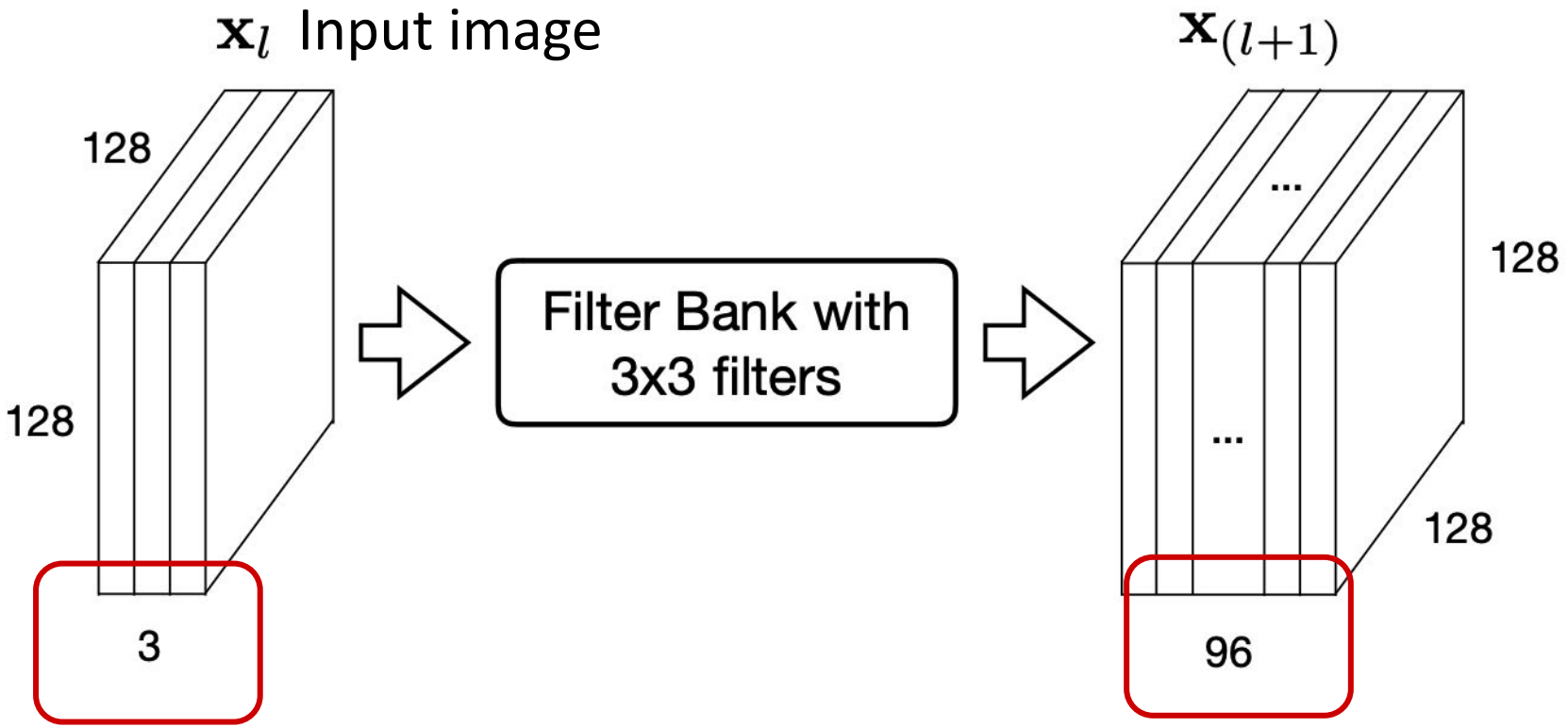
- How many input and output channels are present in this architecture? **1 min** ⏰
  - Enter in slido in the next slide.

How many input and output channels are present?

How many input and output channels are present?



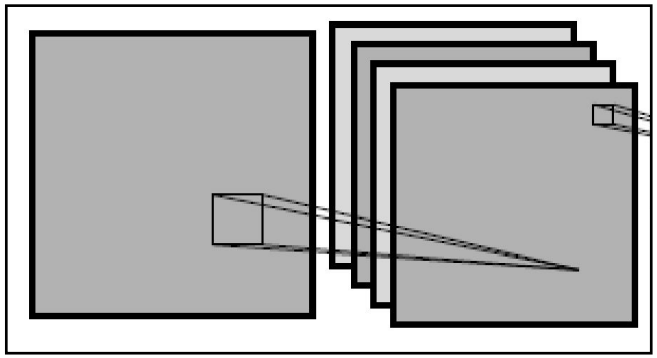
Multiple channels: Example



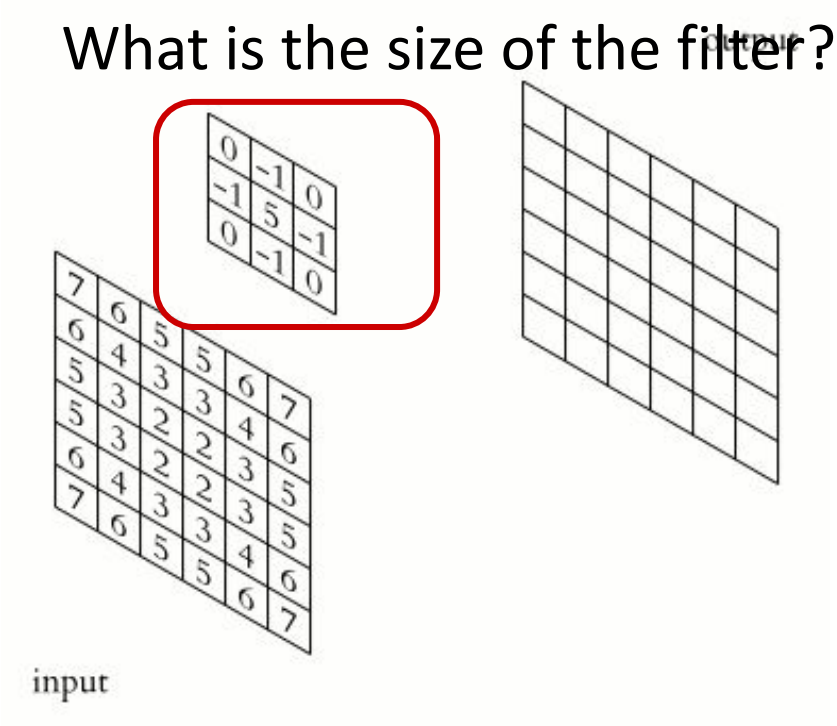
$$\mathbf{x}_l \in \mathbb{R}^{H \times W \times C_l} \quad \longrightarrow \quad \mathbf{x}_{(l+1)} \in \mathbb{R}^{H \times W \times C_{(l+1)}}$$

Multiple channels

- Multiple channels == applying a collection of convolution filters
- Each filter is of a fixed size: M X N
  - 3X3, 7X7



What is the size of the filter?



Terminology so far

Convolution Filter	**M X N**
Channels	Stack of convolutional filters **C**
Feature maps	Outputs after convolution operation applied at a layer

Remember: We are learning all of these during forward and backward pass!

How many parameters are being learnt per layer?

How many parameters are being learnt per layer?

Filter sizes

When mapping from

$$\mathbf{x}_l \in \mathbb{R}^{H \times W \times C_l} \quad \rightarrow \quad \mathbf{x}_{(l+1)} \in \mathbb{R}^{H \times W \times C_{(l+1)}}$$

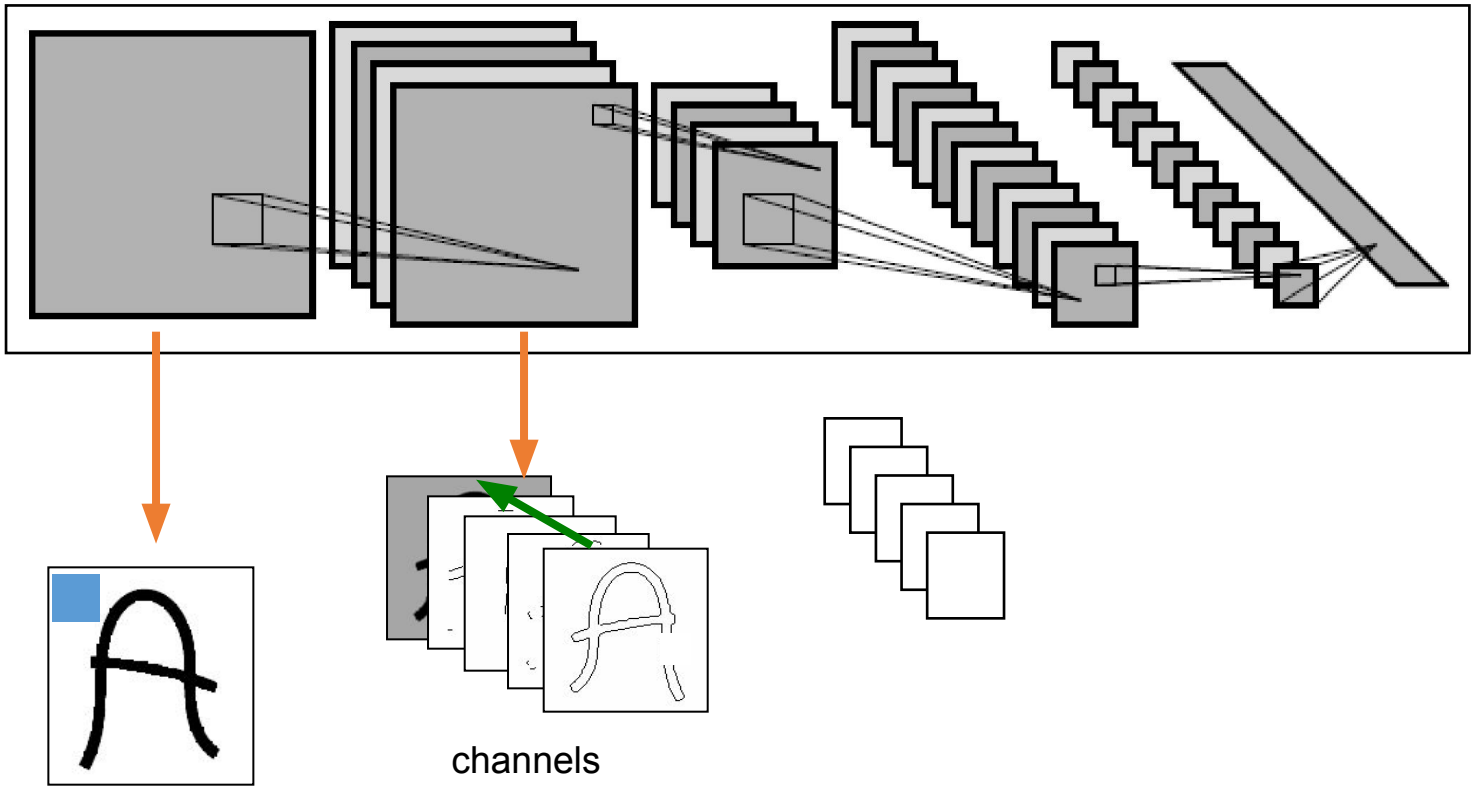
using an filter of spatial extent $M \times N$

Number of parameters per filter: $M \times N \times C_l$

- A very popular ML interview questions

Stacking convolutional layers

- Each **layer** outputs multi-channel feature maps (like images)
- Next layer learns filters on previous layer's feature maps

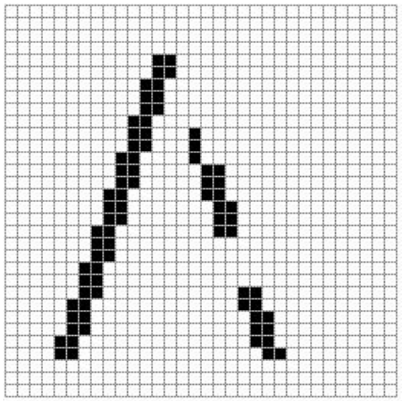


Terminology so far

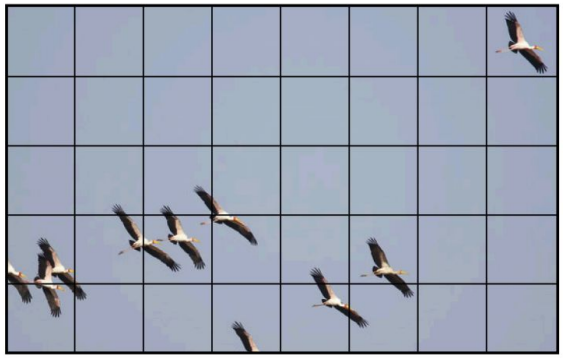
Convolution Filter	**M X N**
Channels	Stack of convolutional filters **C**
Feature maps	Outputs after convolution operation applied at a layer
Layers	Stack of feature maps

Remember: We are learning all of these during forward and backward pass!

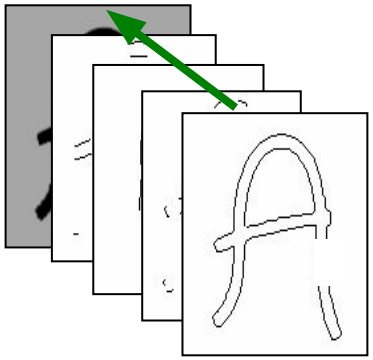
Let’s go back to input representation



Pixel based



Cell based



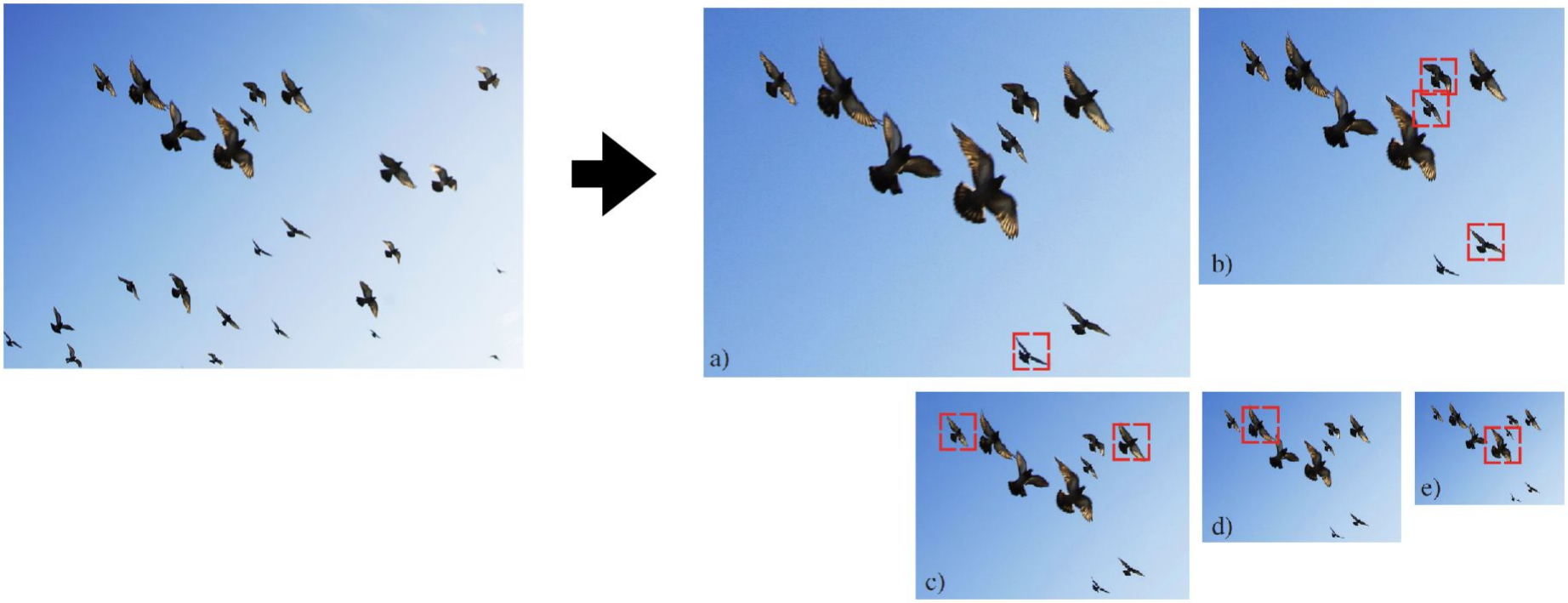
Conv based

- **Advantage of convolutional based representation:**
  - Offers translational invariance.
- What about scale invariance?

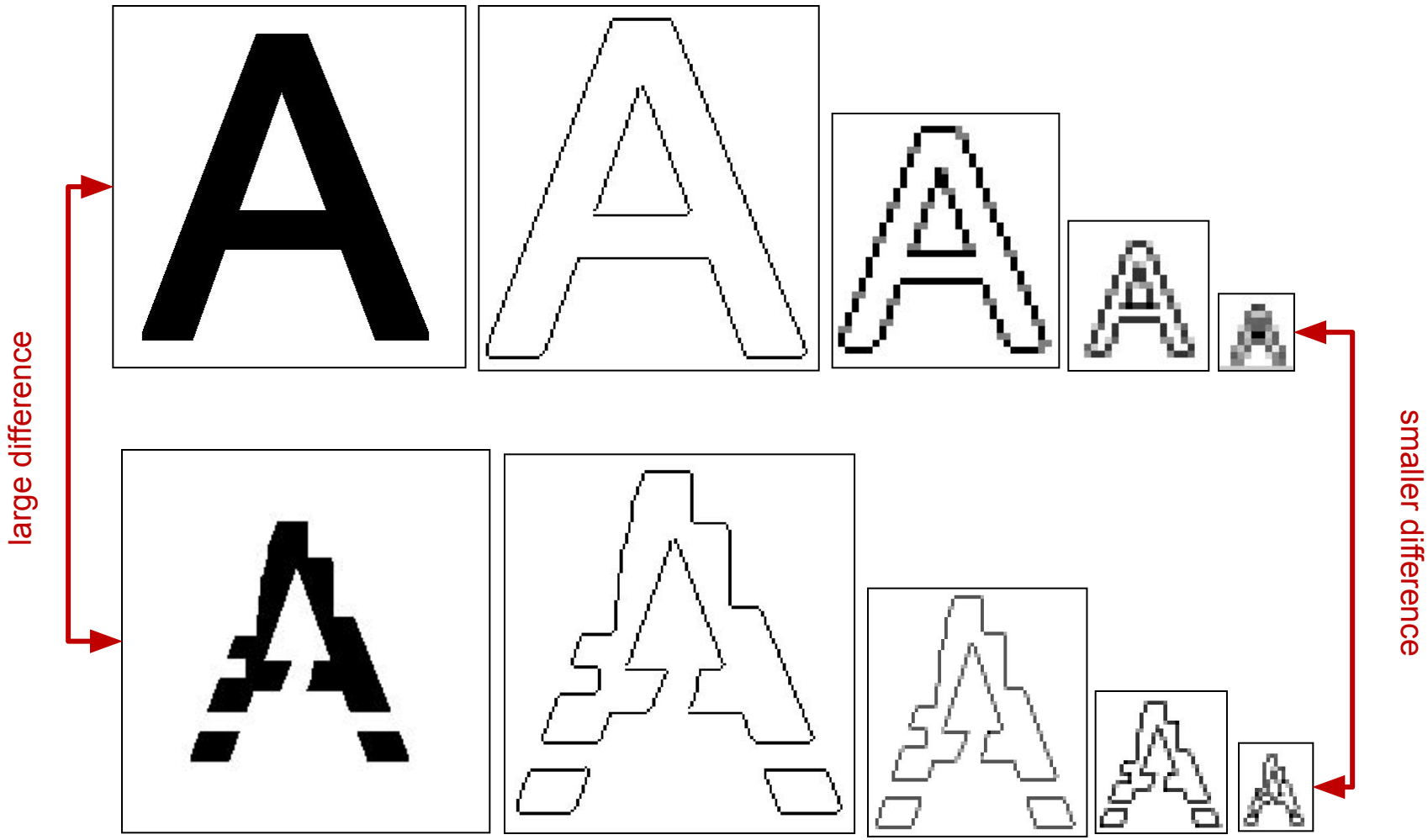
Pooling and downsampling

We need translation and **scale** invariance

Image pyramids

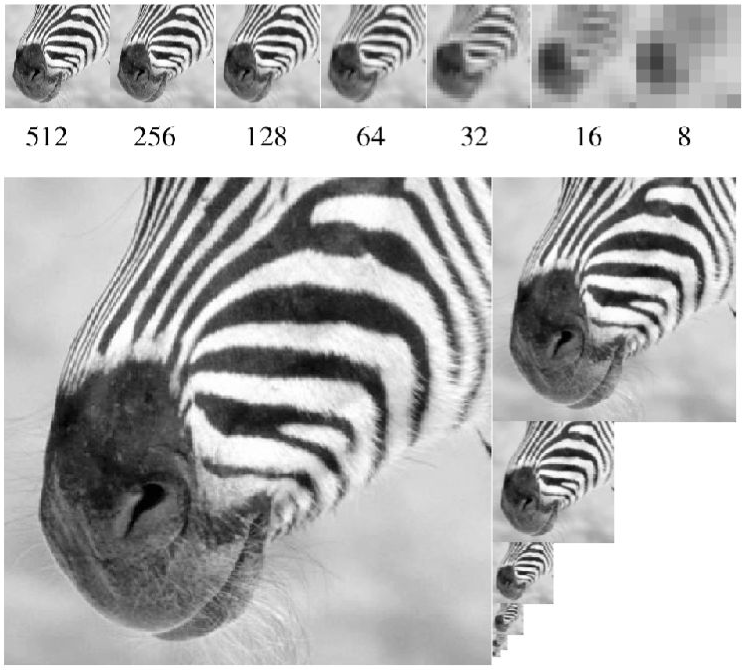


Distortion invariance

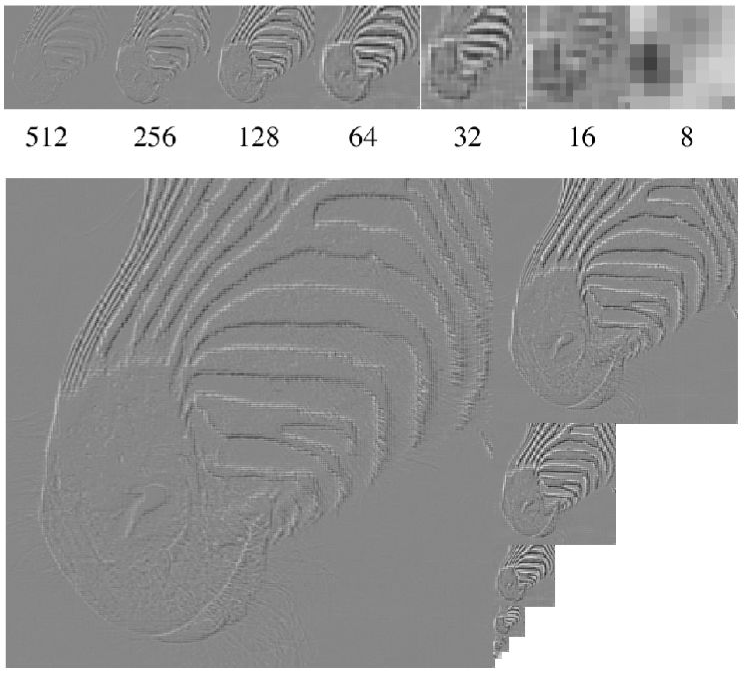


- **Goal:** We want a model that is not fussy (hence robust)

Multiscale representations are great!



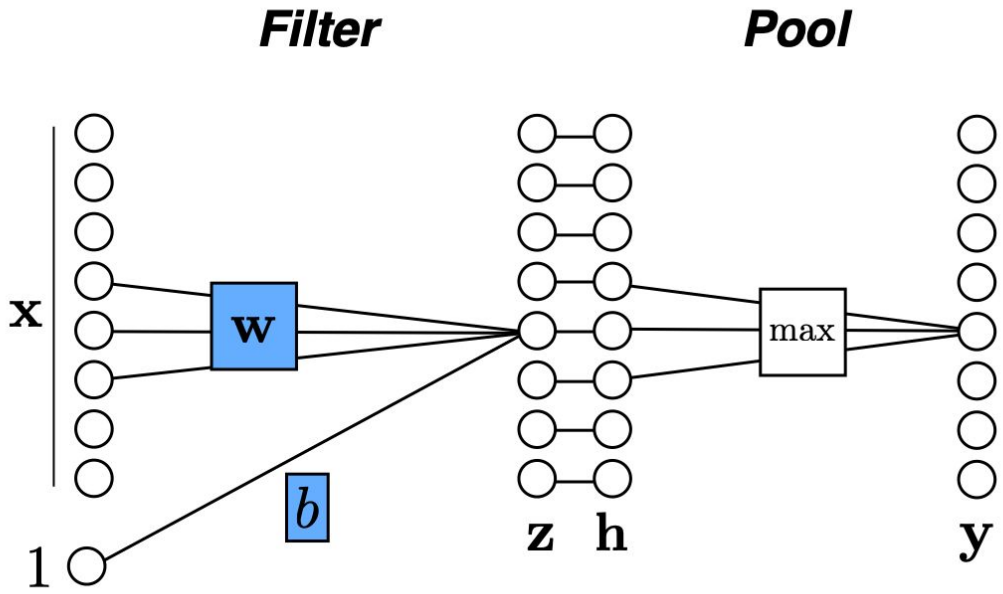
Gaussian Pyr



Laplacian Pyr

How can we use multi-scale modeling in Convnets?

Pooling



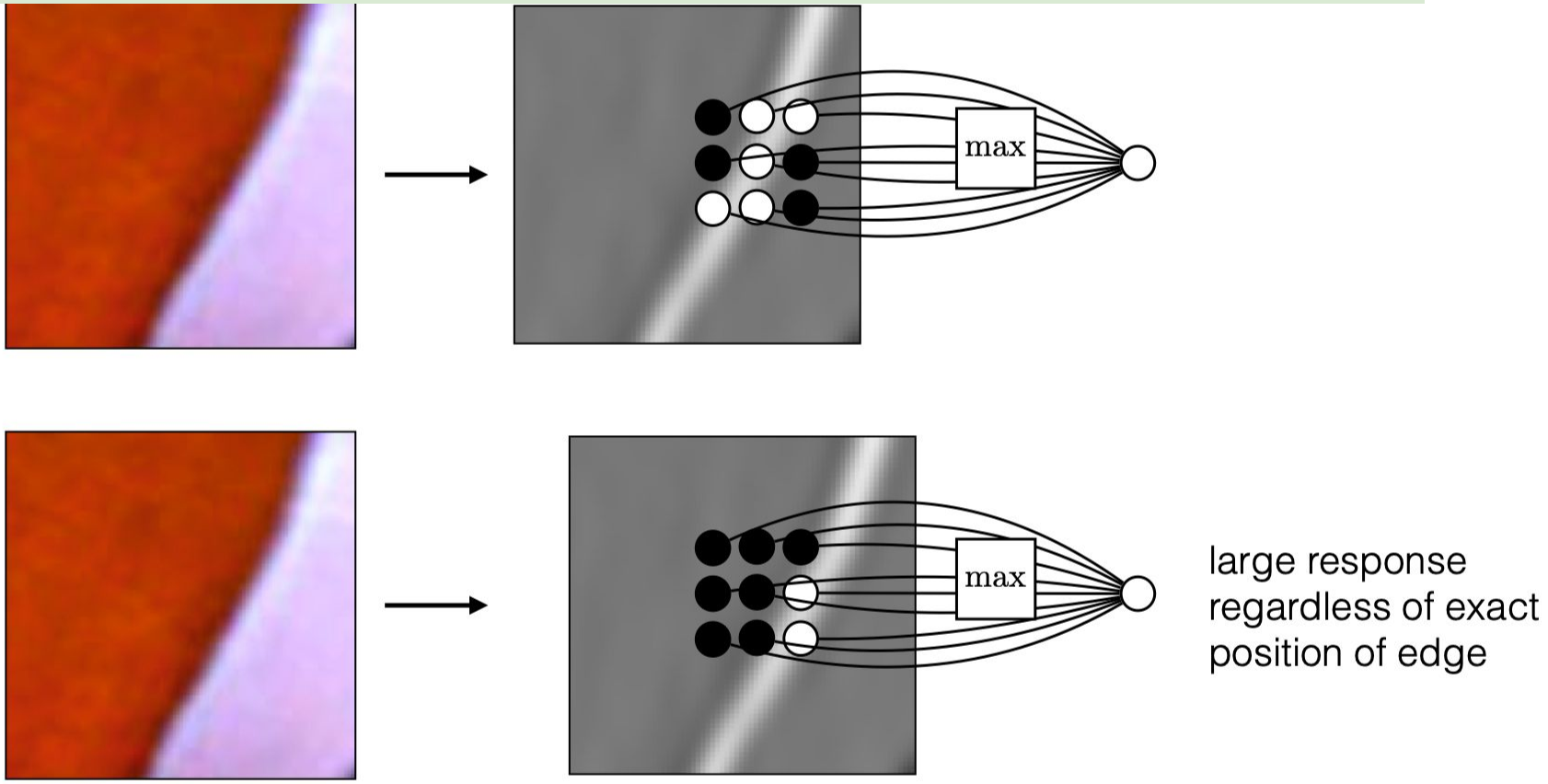
Max pooling

$$y_k = \frac{1}{|\mathcal{N}|} \max_{j \in \mathcal{N}(j)} h_j$$

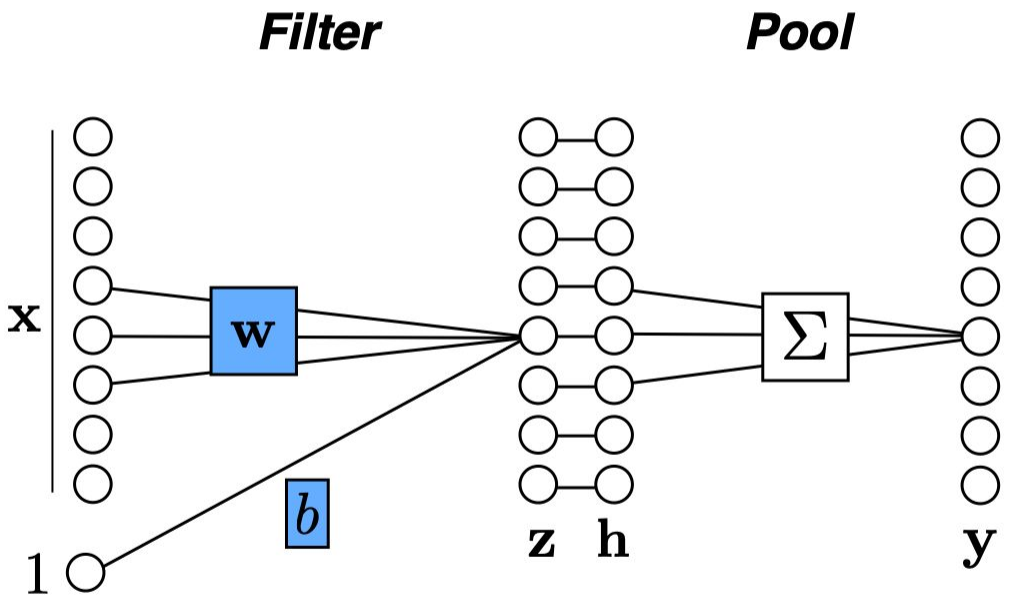
Pooling — Why?

Pooling across spatial locations achieves stability w.r.t. small translations:

- **Goal:** We want a model that is not fussy (hence robust)



Pooling



Max pooling

$$y_k = \frac{1}{|\mathcal{N}|} \max_{j \in \mathcal{N}(j)} h_j$$

Mean pooling

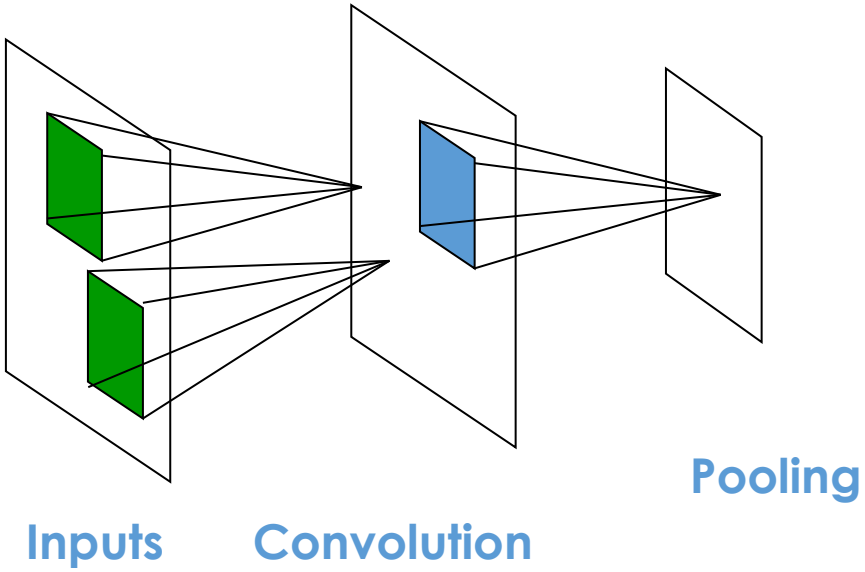
$$y_k = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}(j)} h_j$$

CNNs are stable w.r.t. diffeomorphisms

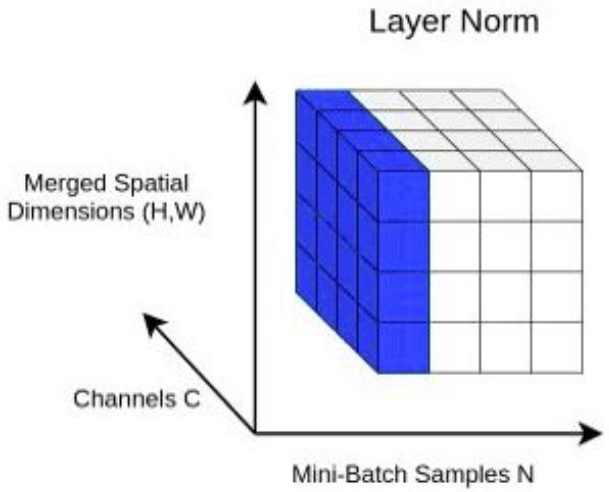
[“Unreasonable effectiveness of Deep Features as a Perceptual Metric”, Zhang et al. 2018]

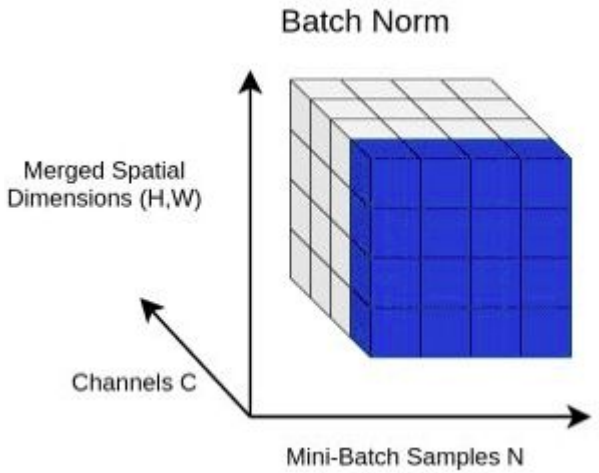
Spatial downsampling

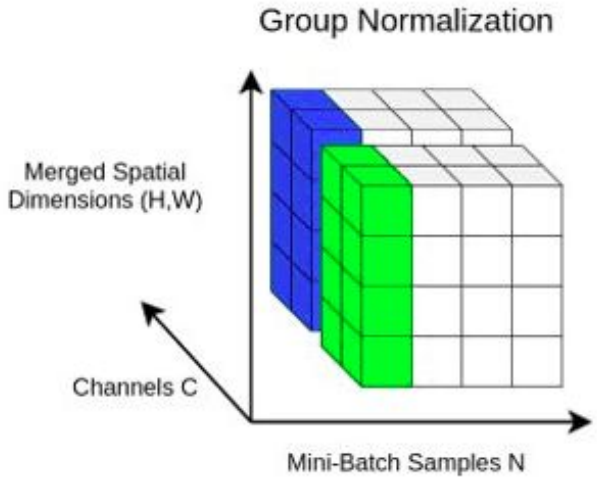
- Offers a way to downsample the feature map
- Makes training computationally tractable.



Recall: Different forms of normalizations





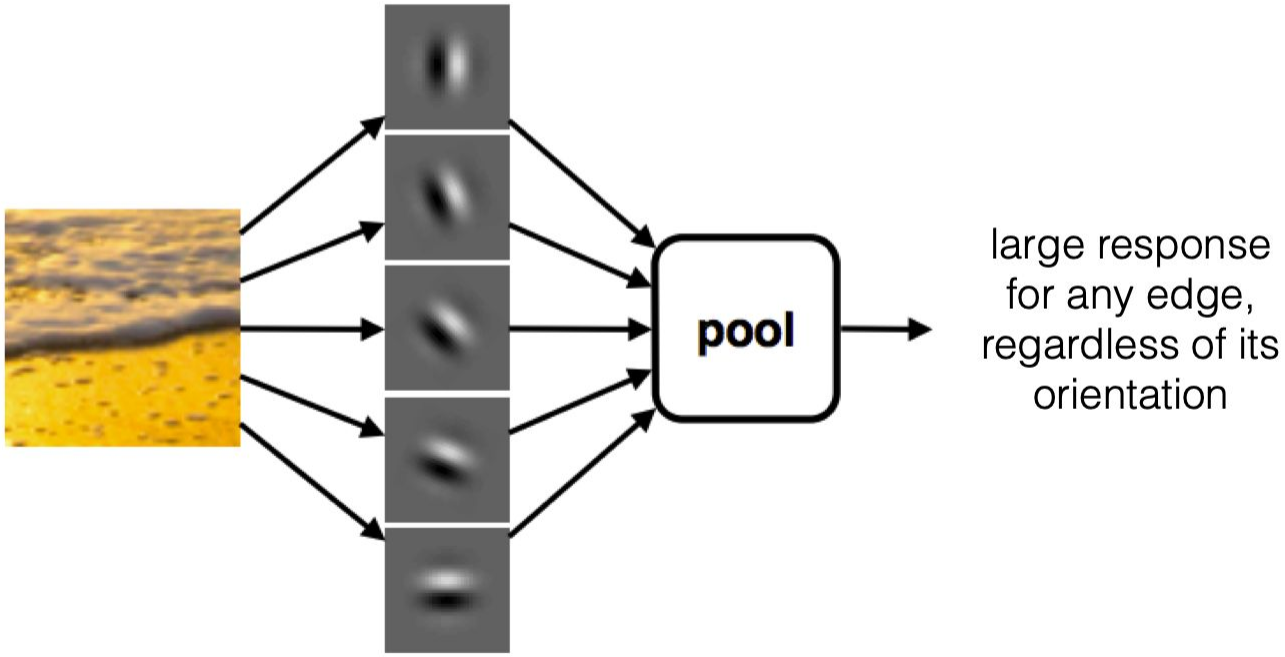


Types of pooling

- Spatial
  - Pool within a given feature map in a given channel
- Channel-wise pooling
  - Pooling across channels.

Pooling *across channels* — Why?

Pooling across feature channels (filter outputs)
can achieve other kinds of invariances:



[Derived from slide by Andrea Vedaldi]

Summary: Pooling

- Pooling
  - Offers subtle distortion invariances
  - Offers numerical stability.
  - Makes training computationally tractable.

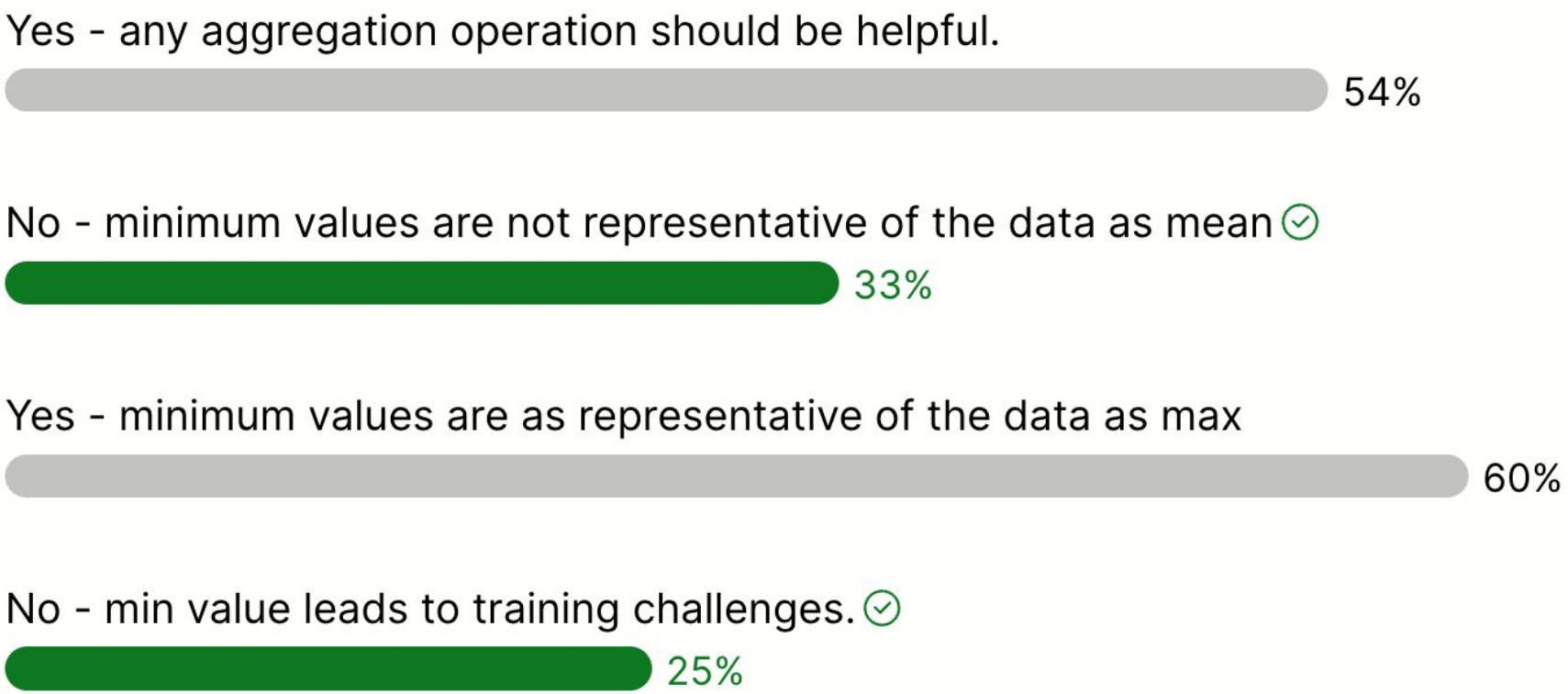
Terminology so far

Convolution Filter	**M X N**
Channels	Stack of convolutional filters **C**
Feature maps	Outputs after convolution operation applied at a layer
Layers	Stack of feature maps
Pooling (nothing to learn!)	Max, mean - across channels or within a feature map

Remember: We are learning all of these during forward and backward pass!

Can we also use min pooling in addition to max and mean pooling? Select all that apply

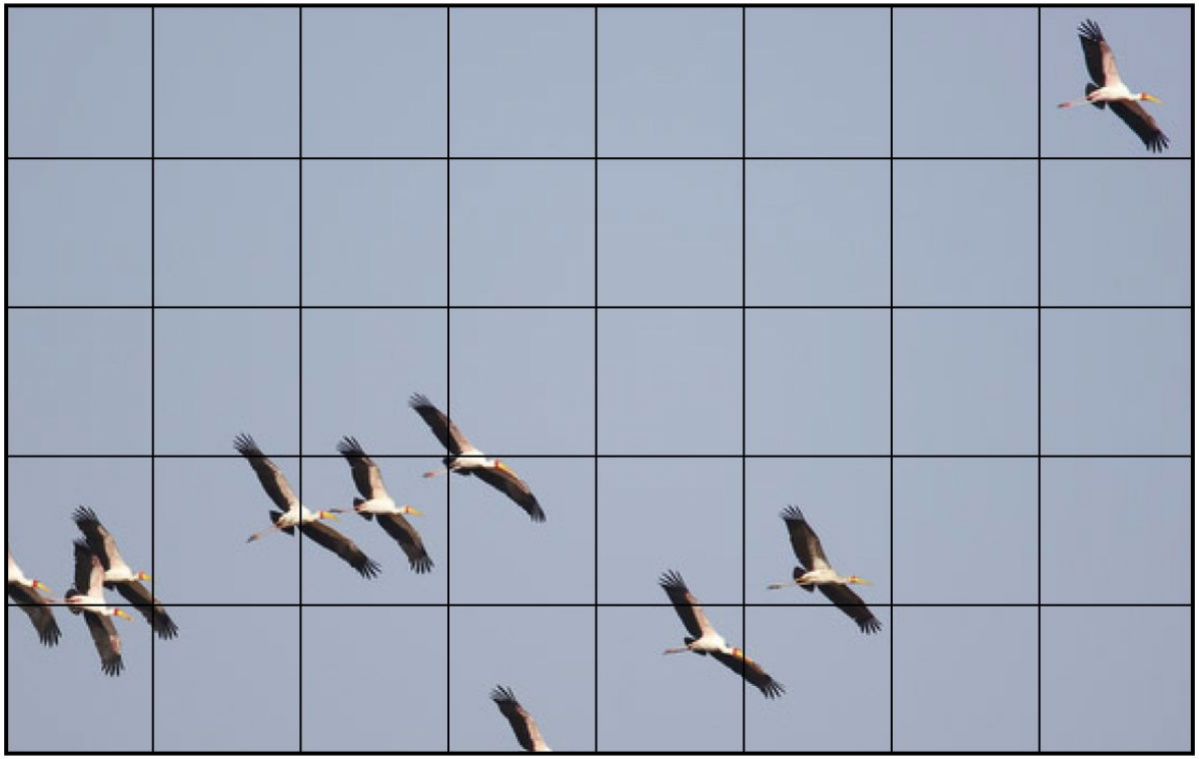
Can we also use min pooling in addition to max and mean pooling? Select all that apply



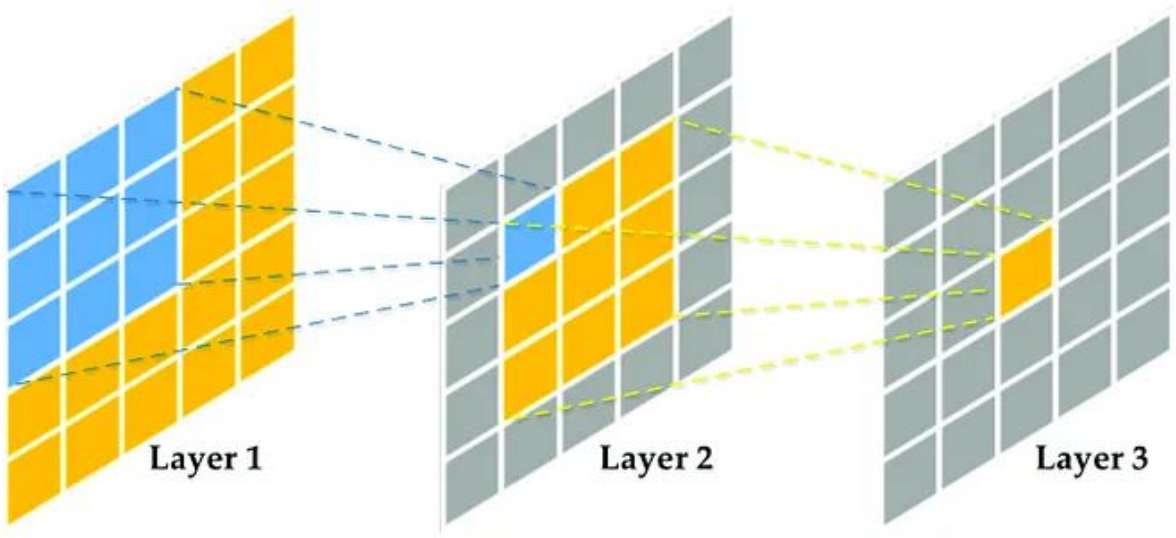
Vanishing gradients

- Min pooling can suppress a lot of signal
- Can lead to gradients being set to 0 during backprop

Receptive fields



Receptive field



- The size of the region in the input that produces the feature.
- Measure of association of an output feature (of any layer) to the input **region** (patch).

Putting it all together…