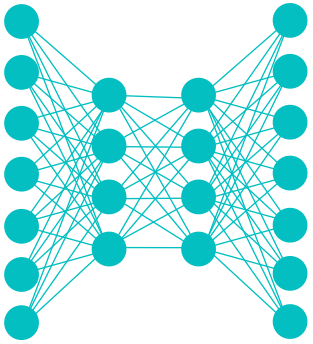
Lecture Notes for

Neural Networks and Machine Learning

CNN Visualization

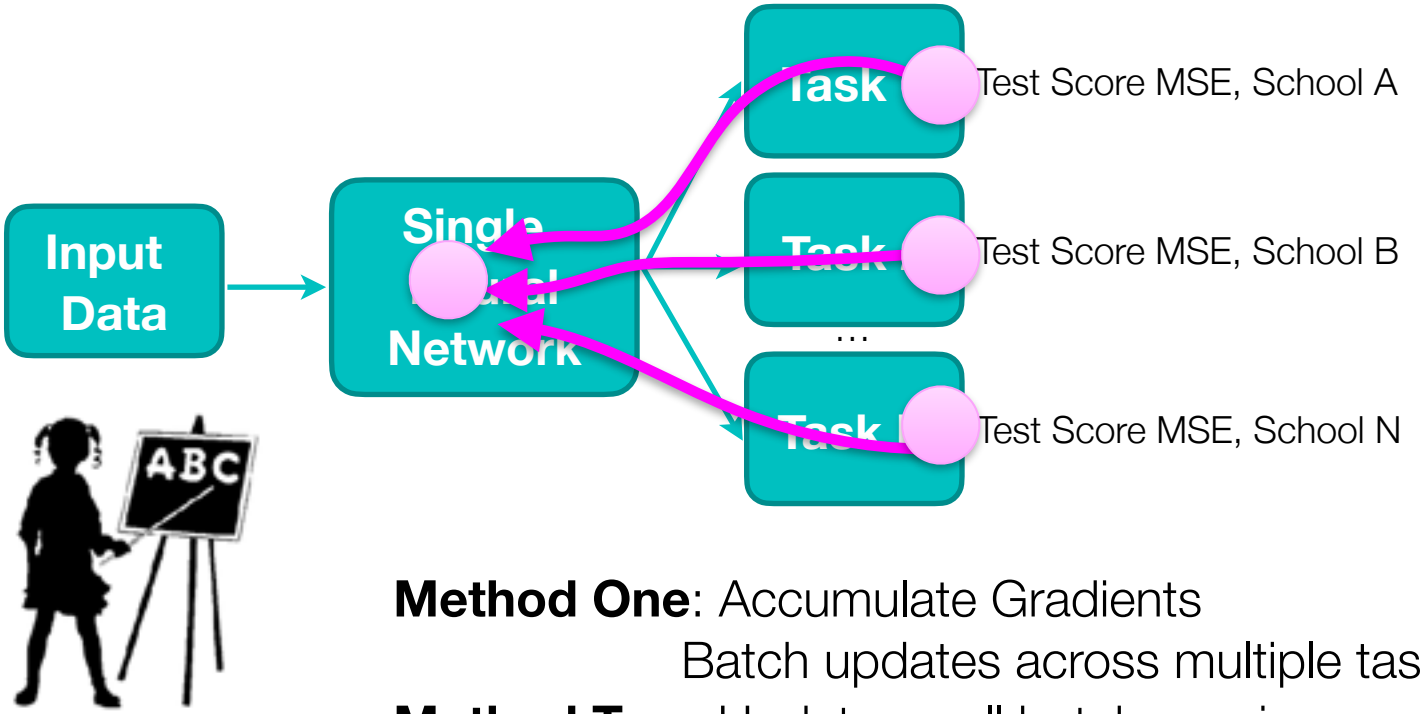


Logistics and Agenda

- Logistics
  - None
- Agenda
  - Finish Multi-Task Demo
  - Visualizing Convolutional Architectures and Demo
- Next Week:
  - Circuits in CNNs
  - Student Paper Presentation: Transformer Interpretability

Multi-task Optimization

Single Task Label per Input



Method One: Accumulate Gradients
Batch updates across multiple tasks

Method Two: Update small batches using a random task
easier, but can cause instability in training

An alternative: Task-Gating

Input Data

Neural Network

Multiply

Classify

Test Score MSE

gating

Task Data

School A
School B

Embed

sigmoid

One Hot

Why is this approach similar to having multi-task outputs?

Multi-Task Learning

School Data, Computer Surveys

Sponsor Follow

LukeWood Luke Wood

KerasCV Author, Full Time Keras team member & Machine Learning researcher @ Google, Part Time UCSD Ph.D student

PRO

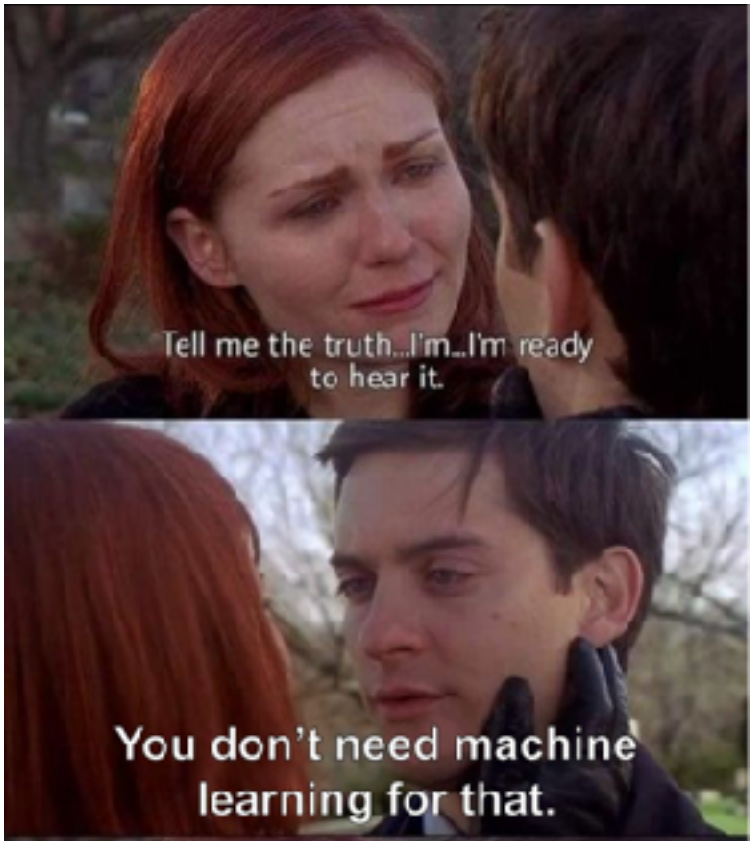
Traian-Pop Traian Pop

Method One: Batch updates across multiple tasks
need to perform customized gradient calculations

Method Two: Update small batches using a random task
easier, but can cause instability in training

`Follow Along: LectureNotesMaster/03 LectureMultiTask.ipynb`

Basics of Convolutional Neural Network Visualization

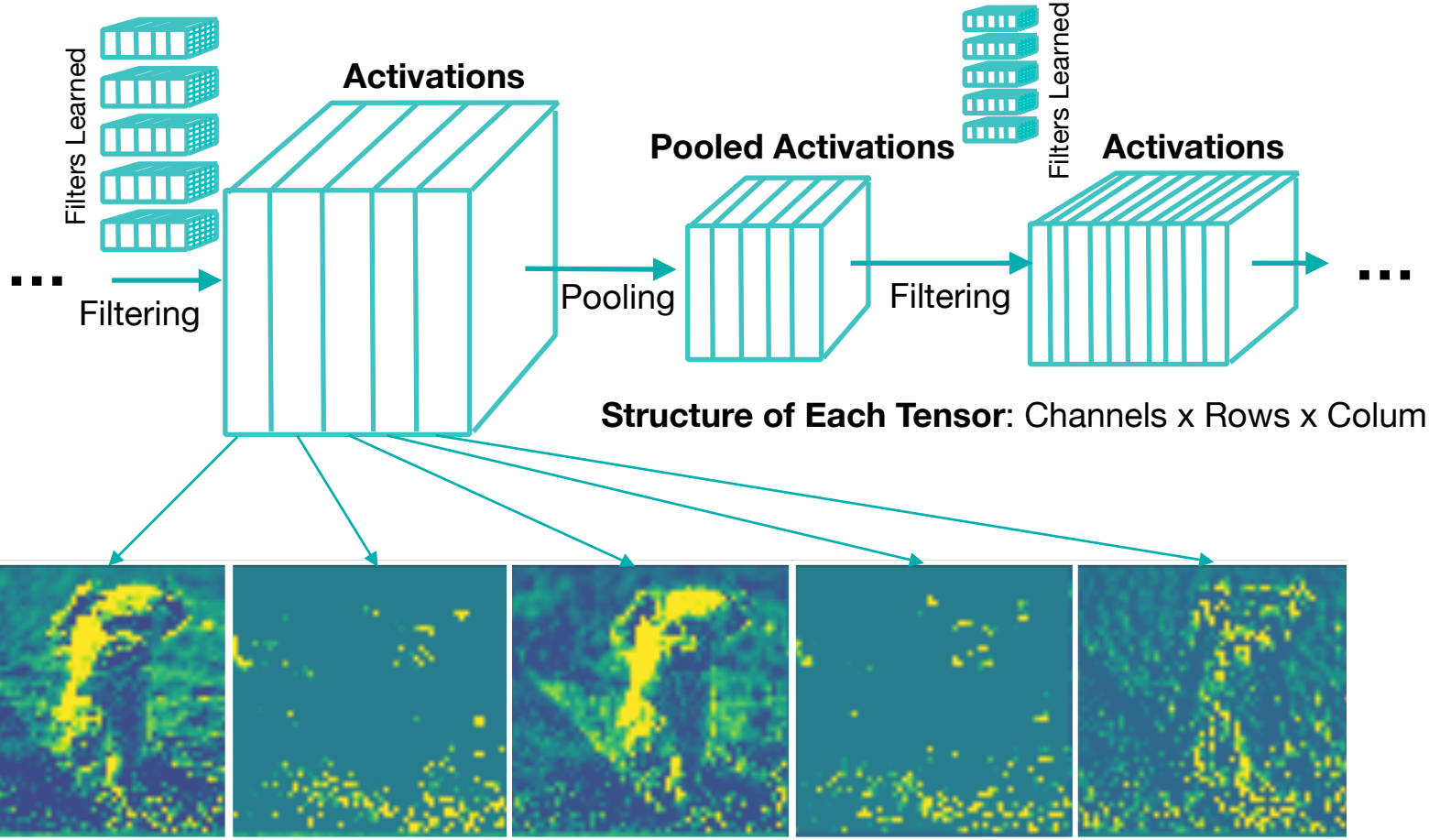


Tools to Visualize Neurons and Filters

- Visualize **Filter Activation**
  - What parts of the inputs activate each filter?
- Visualize **Filters**
  - What does each filter look like? Is it similar to other filters?
  - Can we excite a certain filter by updating the input image?
- **Heatmaps** of Class Activation
  - What part of an input image most influences each final output?

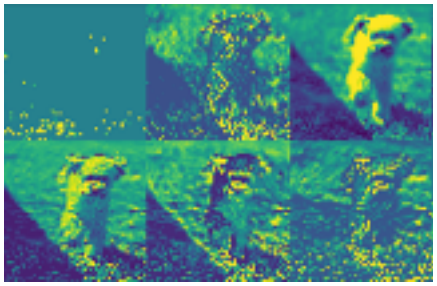
Visualizing Intermediate Activations

- Look layer by layer
- **Assume**: each filter learns something useful



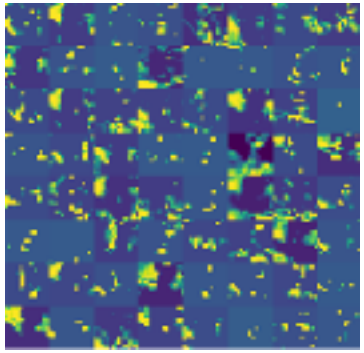
Visualizing Intermediate Activations

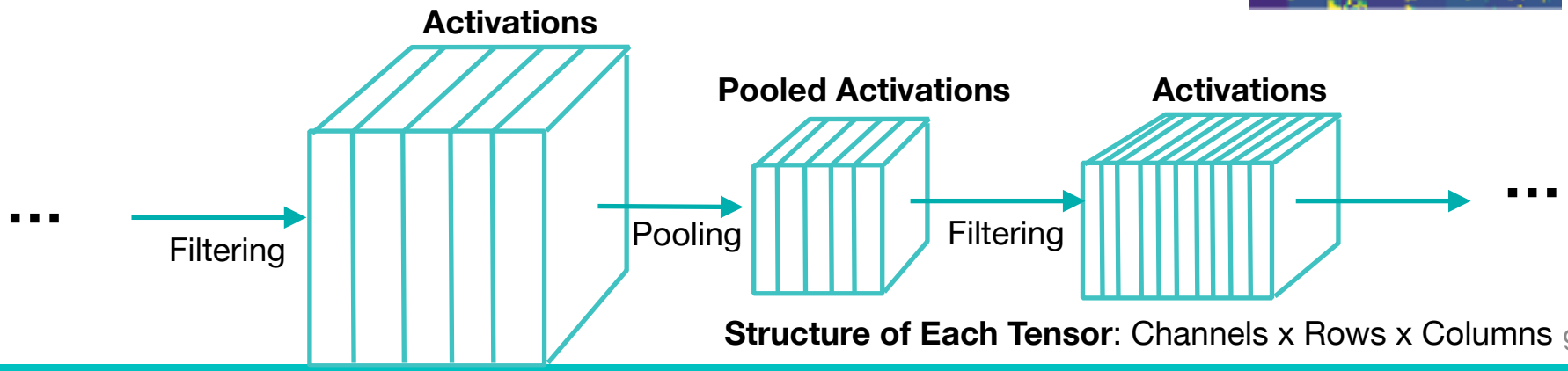
- **Recall**: general structure of most CNNs
  - Small kernels throughout (3x3)
  - Filtering followed by Pooling (spatial downsampling)
  - More filters in later layers



Early Activations are larger but not as numerous

Later Activations are smaller and more numerous

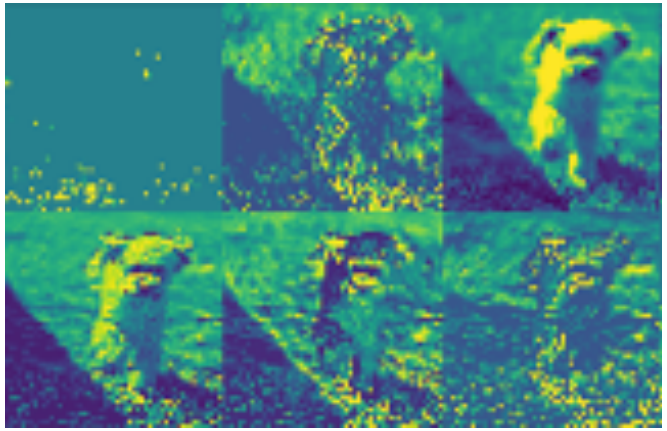




Structure of Each Tensor: Channels x Rows x Columns

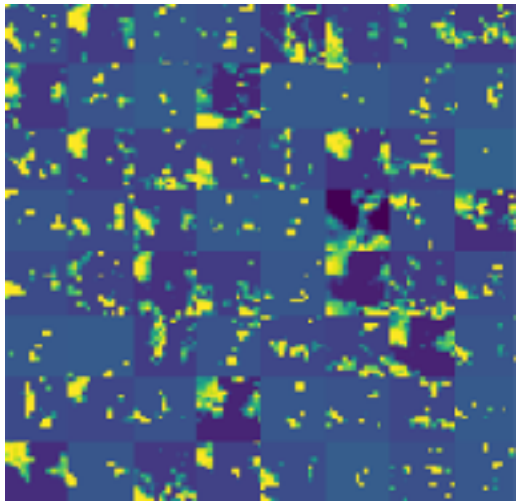
Visualizing Intermediate Activations

- **Result:** Information Distillation Pipeline
  - Deeper layers have more abstract triggers
  - Deeper activations are increasingly sparse
  - Early layers are texture and edge detectors
  - Notion of “High Level Abstraction,” has biological motivation



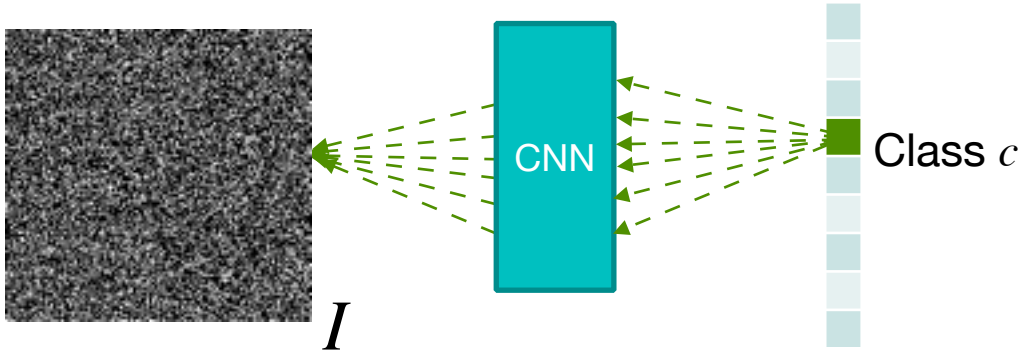
Early Activations are larger but not as numerous

Later Activations are smaller and more numerous

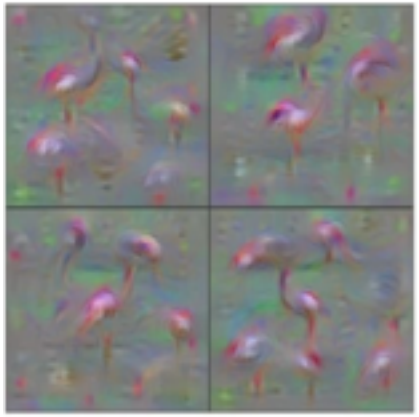


Visualizing Filters: Class Neuron

- **Idea:** What Maximally Activates a Class Output?
  - Gradient Ascent in the Input Space



$$I \leftarrow I + \eta \nabla f_c(I)$$


Flamingo

where c is a specific neuron in output layer

f is the neural network function

I is the input image, init to zeros (or random)

∇ is the gradient of f_c w.r.t I

CNN weights stay unchanged

http://cs231n.github.io/understanding-cnn/

Visualizing Filters: Maximal Activations

Method Two

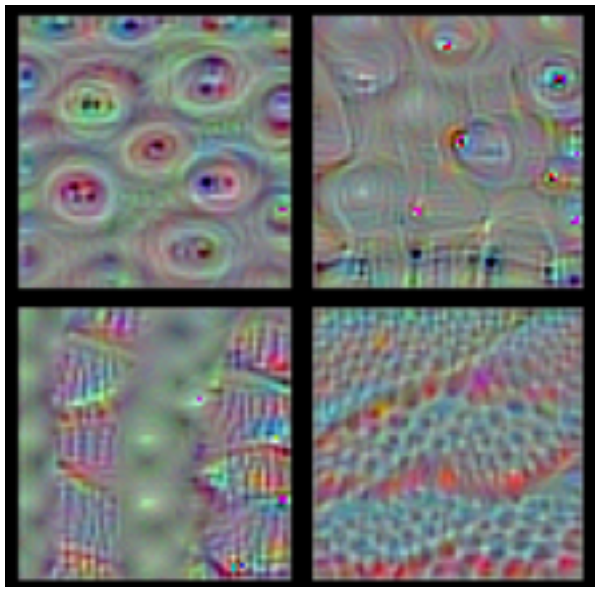
- **Idea:** What Maximally Activates a **Filter**?
  - **Again**: Gradient Ascent in the Input Space

$$I \leftarrow I + \eta \sum_{i,j} \nabla f_n(I)_{i,j}$$

"trick" use norm of gradient

where n is a specific **filter** in a layer

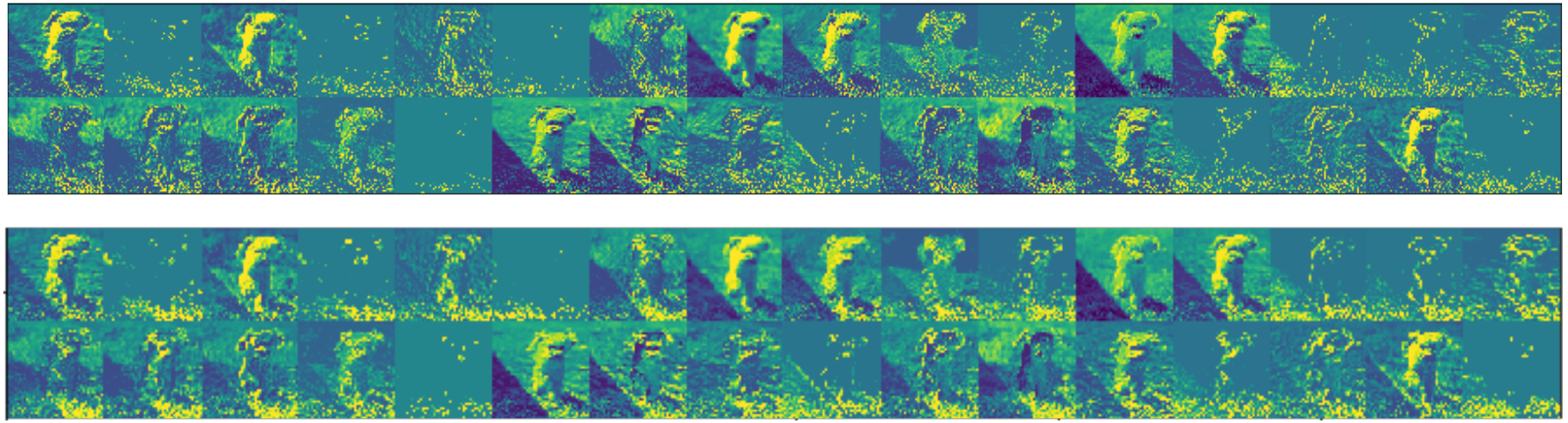
f is the function to n^{th} filter in layer



Visualizing ConvNets

Part One: Filter Activations

Part Two: Image Gradients



```
Follow Along: 04 LectureVisualizingConvnets.ipynb
activation-demo
```