Admin

• Assignment 5 now due tomorrow at 2pm.
  – Extra offices hours today at 4:30-5:30 in ICICS 104.
  – 1 late day to hand in Monday, 2

• Assignment 6:
  – Due next Friday (usual late day policy, assuming phantom “classes”).
  – Neural network code updated to be easier to understand/modify.

• Final:
  – December 12 (8:30am – HEBB 100)
  – Covers Assignments 1-6.
  – List of topics will be posted this weekend.
  – Final from last year will be posted next weekend.
  – Closed-book, cheat sheet: 4-pages each double-sided.
Last Time: Convolutions

• **1D convolution:**
  
  – Takes signal ‘x’ and filter ‘w’ to produces vector ‘z’:

  \[ x \ast w = z \]

  – Can be written as a matrix multiplication:

  \[
  W_x = \begin{bmatrix}
  1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
  0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
  0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1
  \end{bmatrix} = z
  \]
Last Time: Convolutions

• 2D convolution:
  – Signal ‘x’, filter ‘w’, and output ‘z’ are now all images/matrices:

\[
x \ast w = z
\]

  – Vectorized ‘z’ can be written as a matrix multiplication with vectorized ‘x’:

\[
W = \begin{bmatrix}
-2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
\end{bmatrix}
\]
Last Time: Convolutional Neural Networks

- Classic approach uses **fixed convolutions** as features:
  - Usually have **different types/variances/orientations**.
  - Can do subsampling or taking **maxes across locations/orientations/scales**.
Last Time: Convolutional Neural Networks

- Convolutional neural networks learn the features:
  - Learning ‘W’ and ‘w’ automatically chooses types/variances/orientations.
  - Can do multiple layers of convolution to get deep hierarchical features.

Last Time: Convolutional Neural Networks

- Classic **convolutional neural network** (LeNet):

- Visualizing the “activations” of the layers:
  - [http://scs.ryerson.ca/~aharley/vis/conv](http://scs.ryerson.ca/~aharley/vis/conv)
AlexNet Convolutional Neural Network

- ImageNet 2012 won by **AlexNet**:
  - 15.4% error vs. 26.2% for closest competitor.
  - 5 convolutional layers.
  - 3 fully-connected layers.
  - SG with momentum.
  - ReLU non-linear functions.
  - Data translation/reflection/cropping.
  - L2-regularization + Dropout.
  - 5-6 days on two GPUs.

Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.
AlexNet Convolutional Neural Network

- ImageNet 2012 won by AlexNet:
  - 15.4% error vs. 26.2% for closest competitor.

Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The

Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by $253,440 \rightarrow 186,624 \rightarrow 64,896 \rightarrow 64,896 \rightarrow 43,264 \rightarrow 6 \rightarrow 4096 \rightarrow 1000$. 

ZFNet Convolutional Neural Network

- ImageNet 2013 won by variation of AlexNet called ZF Net:
  - 11.2% error (now using 11x11 instead of 7x7).
  - Introduced deconvolutional networks to visualize what CNNs learn.

ZFNet Convolutional Neural Network

Layer 2

Deconvolution network giving patch that leads to largest response

Patch from data giving largest response
ZFNet Convolutional Neural Network
ZFNet Convolutional Neural Network

Layer 4

Layer 5

ZFNet Convolutional Neural Network

• Looked at how prediction changes if we hide part of the image:

http://cs231n.github.io/understanding-cnn/
VGG Convolutional Neural Network

- Image 2014 “Localization” Task won by a **19-layer VGG** network:
  - 7.3% error for classification (2nd place).
  - Uses **3x3 convolution layers** with stride of 1:
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.
    - Increases number of non-linear ReLU operations.
  - “Deep and simple”: variants of VGG are the most popular CNNs.

https://www.cs.toronto.edu/~frossard/post/vgg16/
GoogLeNet

• Image 2014 classification task won by GoogLeNet:
  – 6.7% errors.
  – 22 layers
    • No fully connected layers.
    • During training, try to predict label at multiple locations.
      – During testing, just take the deepest predictions.
    • “Inception” modules: combine convolutions of different sizes.

ResNet

• Image 2015 won by Resnet (all 5 tasks):
  – 3.6% error (below estimate 5% human error).
  – 152 layers (2-3 weeks on 8 GPUs to train).
  – “Residual learning” allows better performance with deep networks:
    • Include input to layer in addition to non-linear transform.

• Network just focuses on “residual”: what is not captured in original signal.

Mission Accomplished?

• For speech recognition and object detection:
  – No other methods have ever given the current level of performance.
  – But, we also don’t know how to scale up other universal approximators.
  – There is likely some overfitting to these particular tasks.

• Despite high-level of abstraction, deep CNNs are easily fooled:
  – But progress on fixing ‘blind spots’.

http://cs.nyu.edu/~zaremba/docs/understanding.pdf
CNNs for Rating Selfies

Bad selfies

Our training data

Good selfies

https://karpathy.github.io/2015/10/25/selfie/
CNNs for Rating Selfies

Do:
- Be female
- Have face be $\frac{1}{3}$ of image
- Cut off forehead
- Show long hair
- Oversaturate face
- Use filter
- Add border

Don't:
- Use low lighting
- Make head too big
- Take group shots

https://karpathy.github.io/2015/10/25/selfie/
CNNs for Rating Selfies

Finding best image crop:

https://karpathy.github.io/2015/10/25/selfie/
Beyond Classification

• “Fully convolutional” neural networks allow “dense” prediction:
  – Best methods combine these with graphical models or LSTMs (CPSC 540).

• Image segmentation:
Beyond Classification

• “Fully convolutional” neural networks allow “dense” prediction:
  – Best methods combine these with graphical models or LSTMs (CPSC 540).
  
• Depth Estimation:
Beyond Classification

• Image colorization:

- Image Gallery, Video

Inceptionism

• A crazy idea:
  – Instead of weights, use backpropagation to take gradient with respect to $x_i$.

• Inceptionism with trained network:
  – Fix the label $y_i$ (e.g., “banana”).
  – Start with random noise image $x_i$.
  – Use gradient descent on image $x_i$.
  – Add a spatial regularizer on $x_i$:
    • Encourages neighbouring $x_{ij}$ to be similar.

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html
Inceptionism

- Inceptionism for different class labels:
Inceptionism

- **Inceptionism** where we try to match $z_i^{(m)}$ values instead of $y_i$.
  - Shallow ‘m’:
Inceptionism

- **Inceptionism** where we try to match $z_i^{(m)}$ values instead of $y_i$.
  - Deepest ‘m’:
Inceptionism

- **Inceptionism** where we try to match $z_{i}^{(m)}$ values instead of $y_i$.
  - “Deep dream” starts from random noise:

- Inceptionism gallery
- Deep Dream video

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html
Artistic Style Transfer

• **Artistic style transfer:**
  – Given a content image ‘C’ and a style image ‘S’.
  – Make a image that has content of ‘C’ and style of ‘S’.
Artistic Style Transfer

- **Artistic style transfer:**
  - Given a *content image* ‘C’ and a *style image* ‘S’.
  - Make an image that has *content of ‘C*’ and *style of ‘S’*. 

Artistic Style Transfer

• **Artistic style transfer:**
  – Given a content image ‘C’ and a style image ‘S’.
  – Make a image that has content of ‘C’ and style of ‘S’.

• CNN-based approach applies gradient descent with 2 terms:
  – Loss function: *match deep latent representation of content image ‘C’*:
    • Difference between $z_i^{(m)}$ for deepest ‘m’ between $x_i$ and ‘C’.
  – Regularizer: *match all latent representation covariances of style image ‘S’*.
    • Difference between covariance of $z_i^{(m)}$ for all ‘m’ between $x_i$ and ‘C’.
Artistic Style Transfer
Examples

Figure: Left: My friend Grant, Right: Grant as a pizza
Artistic Style Transfer

- Recent methods combine CNNs with graphical models (CPSC 540):

Artistic Style Transfer

- Recent methods combine CNNs with graphical models (CPSC 540):

  ![Input style](https://arxiv.org/pdf/1601.04589.pdf)

  ![Input content](https://arxiv.org/pdf/1601.04589.pdf)

  ![Ours](https://arxiv.org/pdf/1601.04589.pdf)
Artistic Style Transfer for Video

• Combining style transfer with optical flow:
  – https://www.youtube.com/watch?v=Khuj4ASIdmU

• Videos from Ricky’s paper: