Today: Outline

- **Pre-lecture material**

- **Recurrent networks applications**

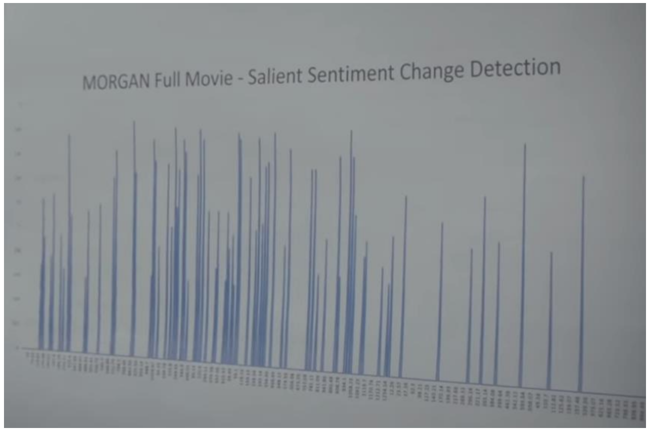
- **NN training strategies**

- **Reminders:** PS3, due Mar 16
  Special Accommodations (if any), by Mar 17

AI Generated Trailer

- Analyze a movie and generate a trailer automatically

- How?
  Detecting salient moments
  e.g. action/emotions



Detecting Salient Regions

- Two sample actions:

Handstand Walking

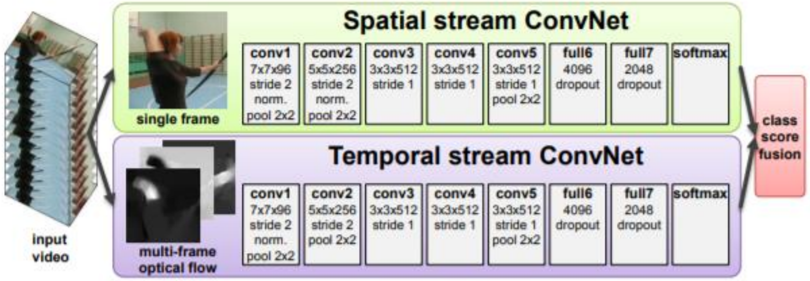
Ice Dancing

AI Generated Match Highlights

- IBM's produce the official match highlights of Wimbledon and US Open tennis tournaments.

- https://www.usopen.org/en_US/video/2017-08-31/1504233424.html

- Multi-modal System



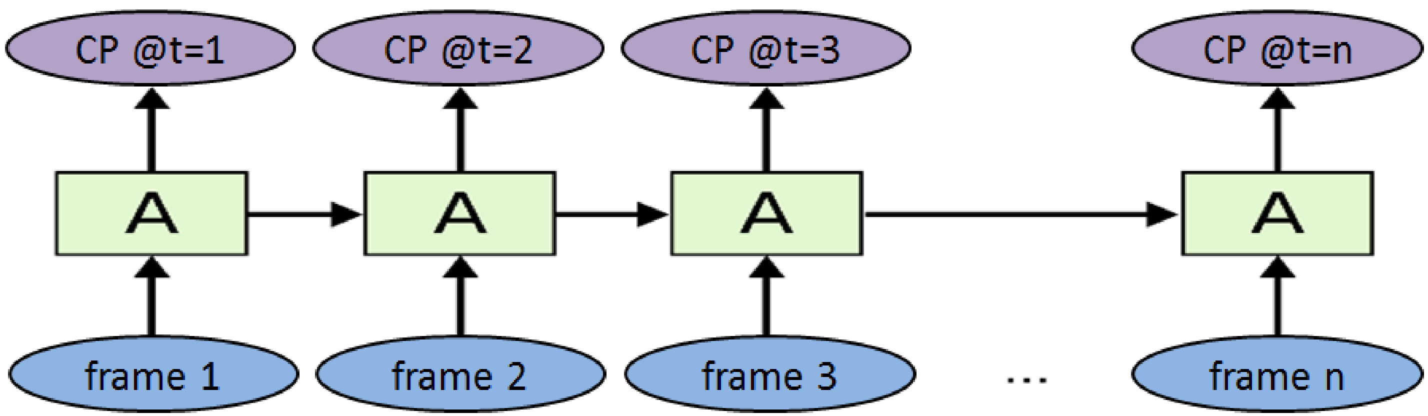
- Bias Considerations

Neural Networks VI

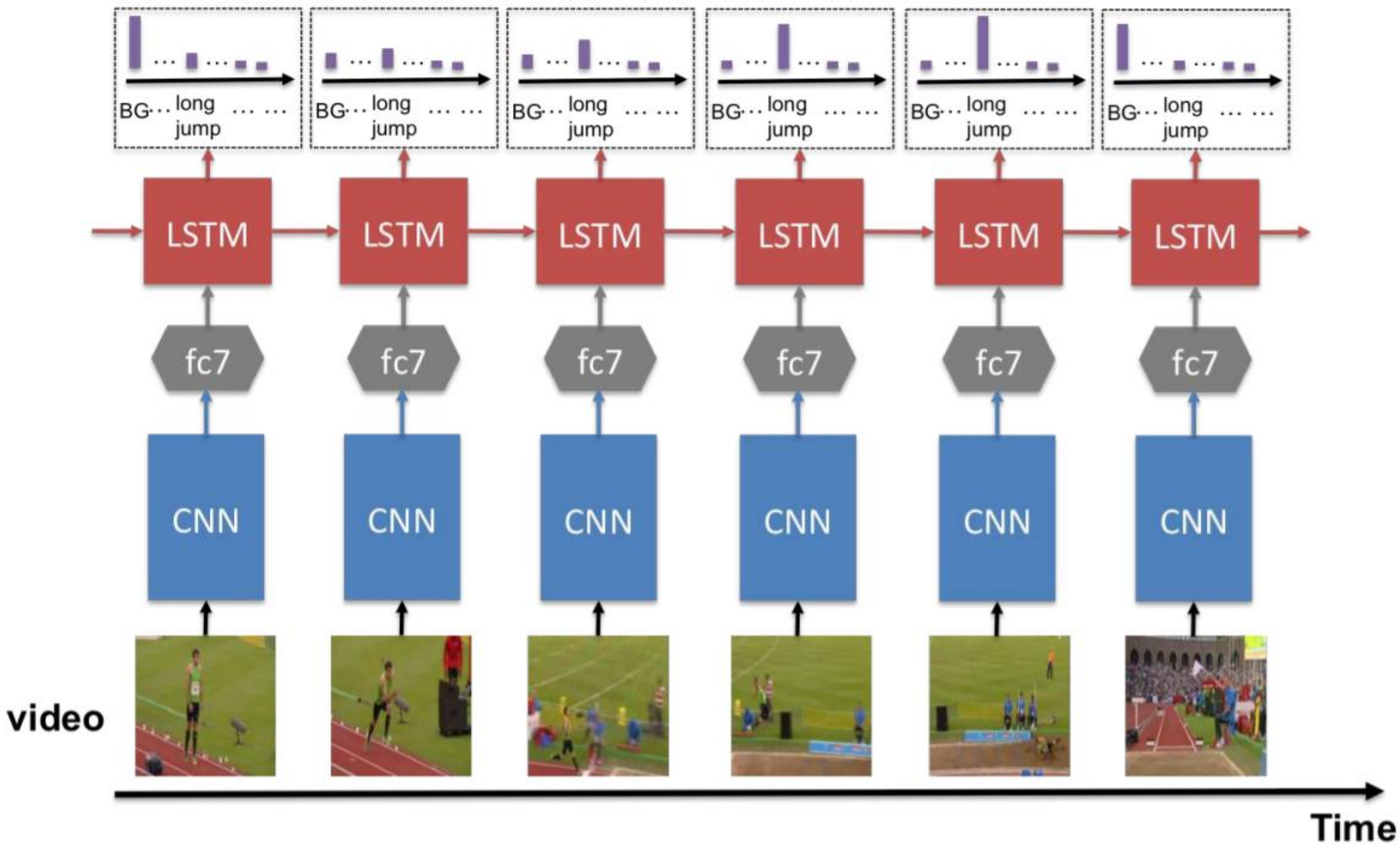
Applications of Recurrent Networks

Application 1: Video Classification

- CP: conditional class probability
- frame i could be a feature describing frame i, example: CNN feature



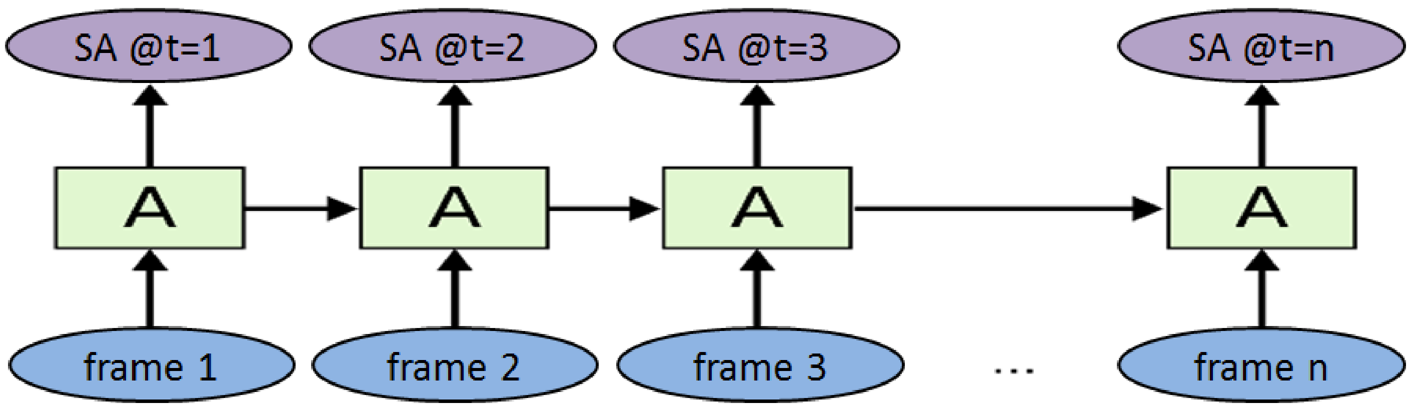
Application 1: Video Classification



Shugao Ma et al., CVPR'16, http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Ma_Learning_Activity_Progression_CVPR_2016_paper.pdf

Application 2: Self-Driving Cars

- SA: steering angle
- frame i could be a feature describing frame i, example: 3D-CNN feature



Application 2: Self-Driving Cars

- DeepTesla

DeepTesla, http://selfdrivingcars.mit.edu/deeptesla/

Application 2: Self-Driving Cars

- Udacity winning team: *Team Komanda*
  - x_t: 3D convolution of image sequence
  - h_t: steering angle, speed, torque

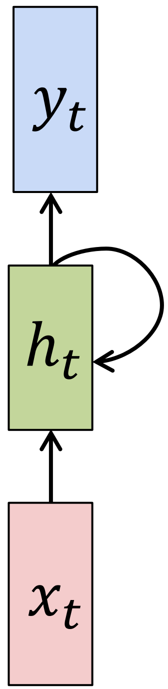
Lex Fridman, lecture, https://www.youtube.com/watch?v=nFTQ7kHQWtc

Application 3: Character RNN

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

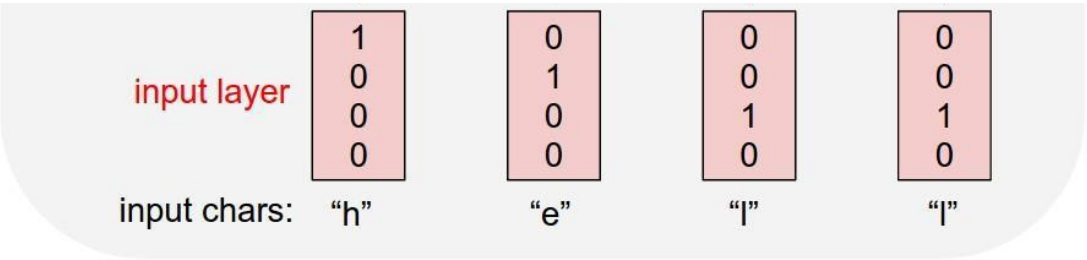


Application 3: Character RNN

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

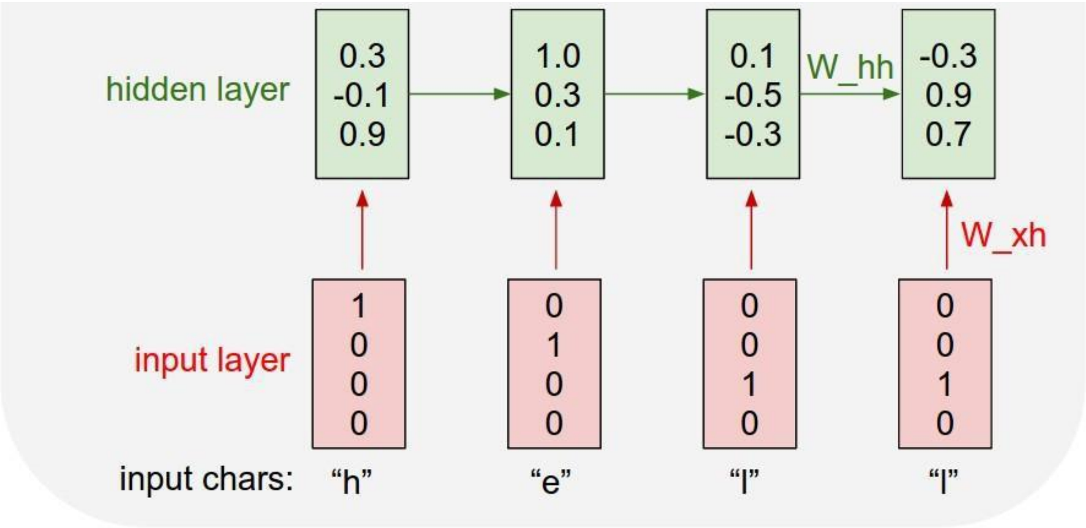


Application 3: Character RNN

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

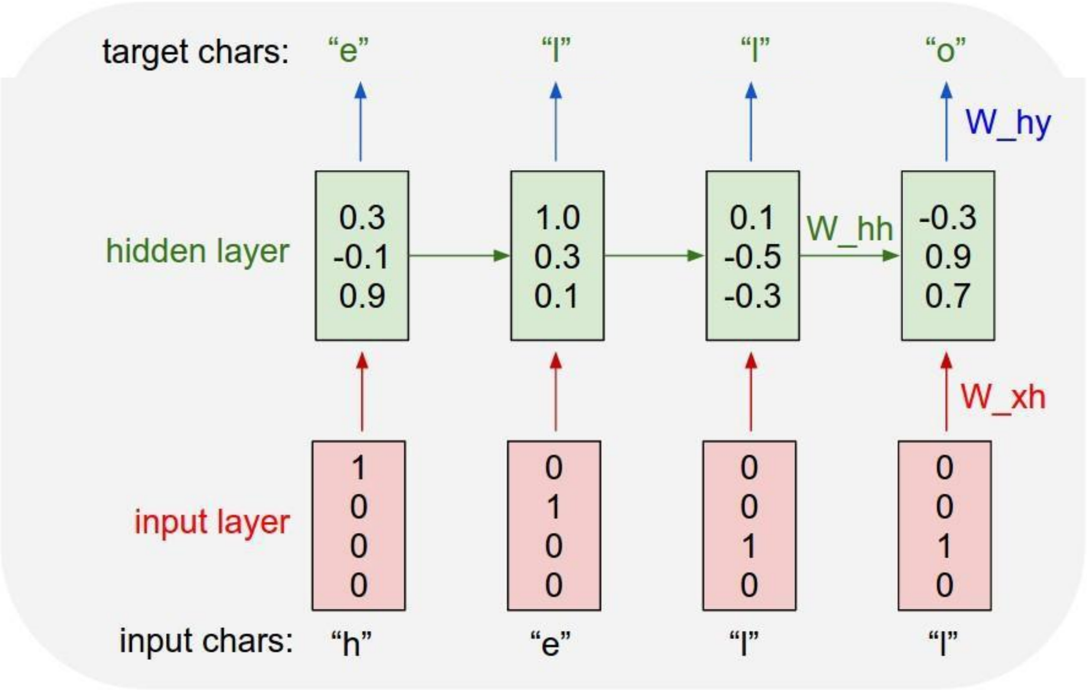


Application 3: Character RNN

Character-level language model example

Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**



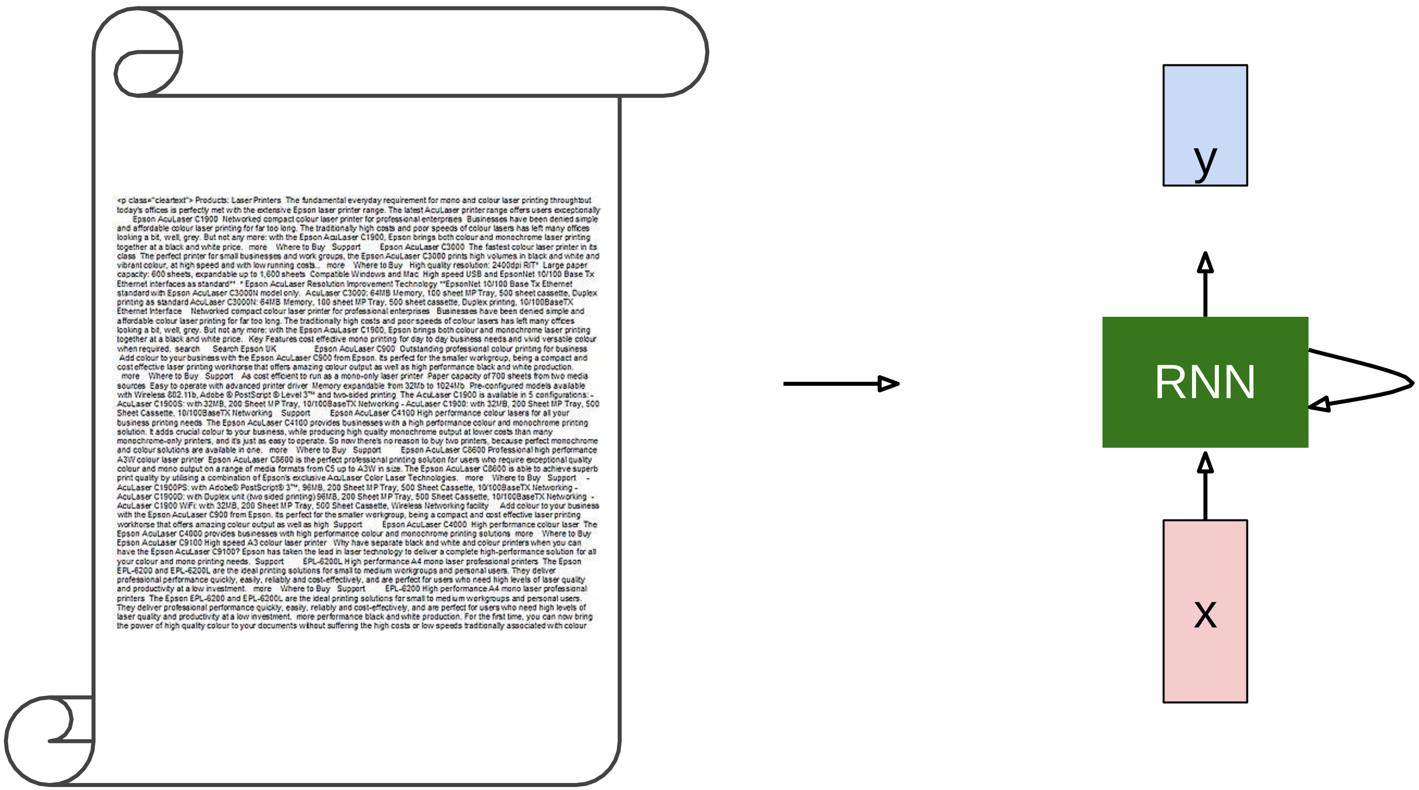
Application 4:Reading cursive handwriting

- This is a natural task for an RNN.
- The input is a sequence of (x,y,p) coordinates of the tip of the pen, where p indicates whether the pen is up or down.
- The output is a sequence of characters.

- Graves & Schmidhuber (2009) showed that RNNs with LSTM are currently the best systems for reading cursive writing.
  - They used a sequence of small images as input rather than pen coordinates.

Application 5: StyleText Generation

Training text: William Shakespeare



Application 5: StyleText Generation

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

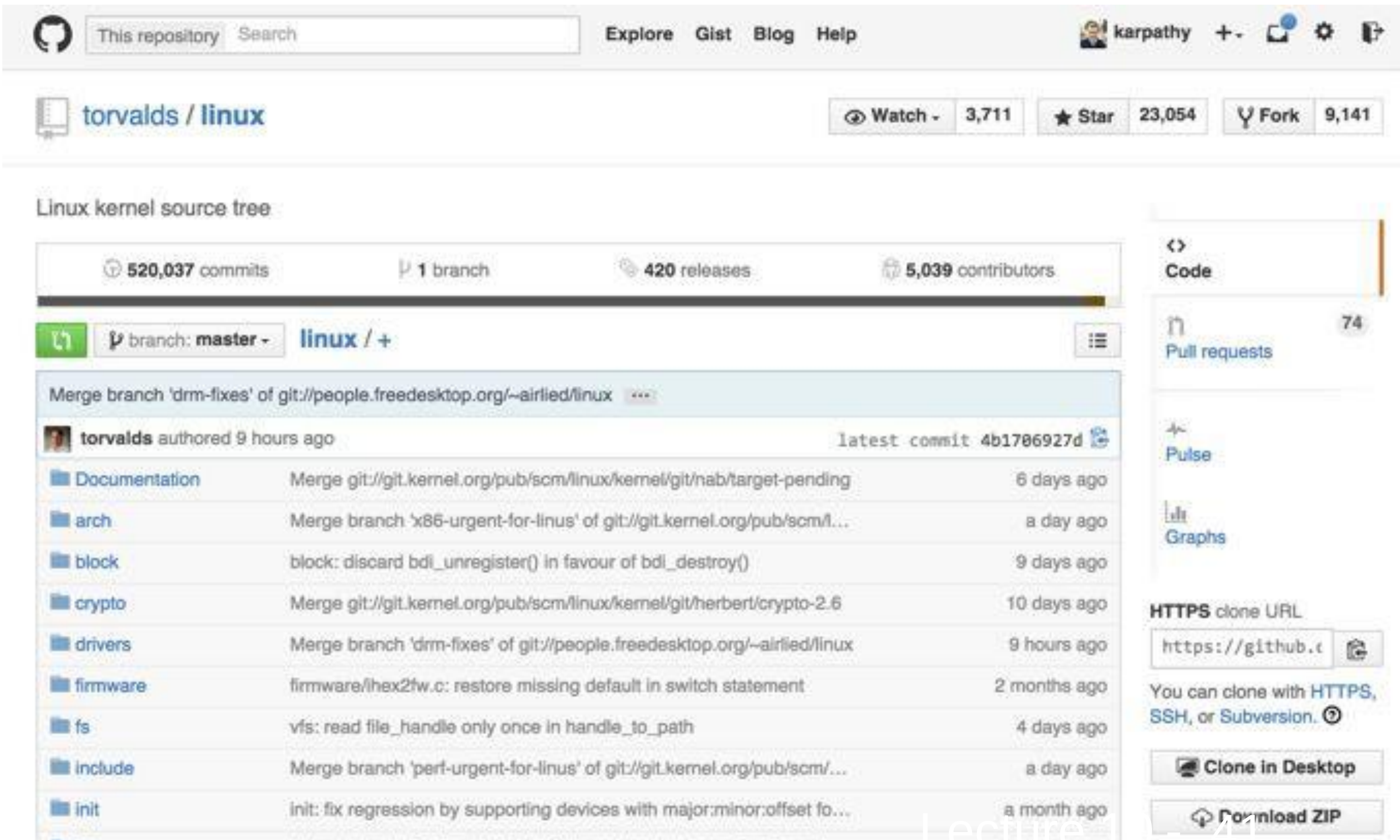
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Application 6: Code Generation

Train on C code



Application 6: Code Generation

```
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

Generated C code

Application 7: Writing a Movie Script

https://arstechnica.com/the-multiverse/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/

Neural Networks VI

Training Strategies

Universality

- Why study neural networks in general?
  - Neural network can approximate any continuous function, even with a single hidden layer!
  - http://neuralnetworksanddeeplearning.com/chap4.html

Why Study Deep Networks?

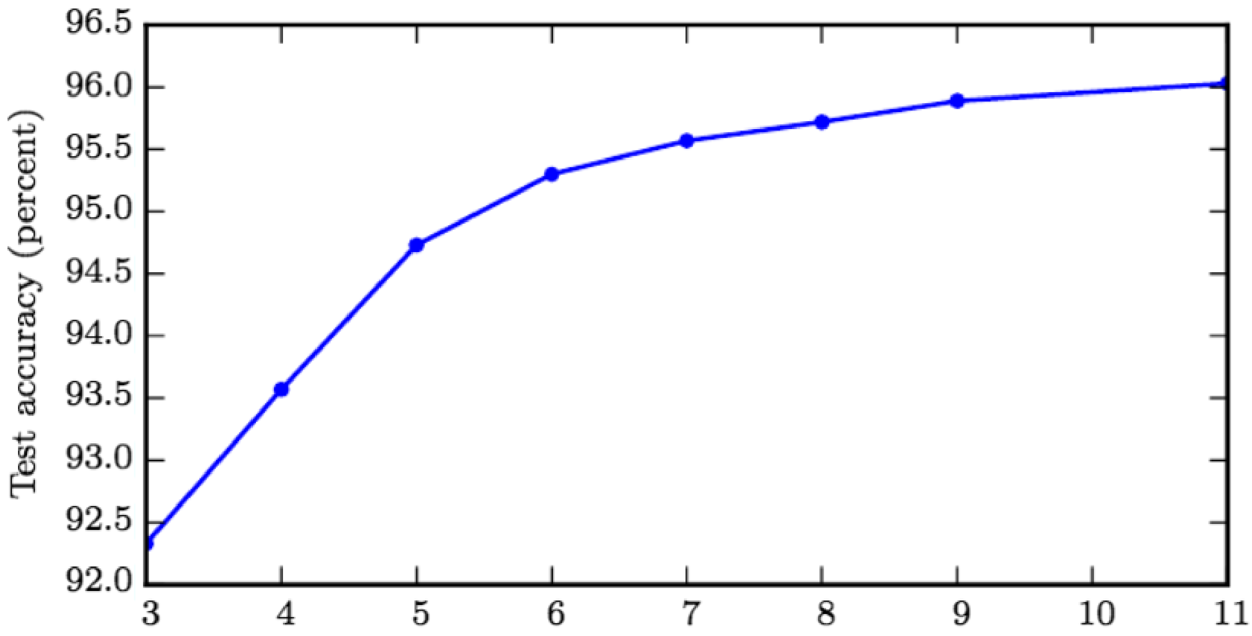
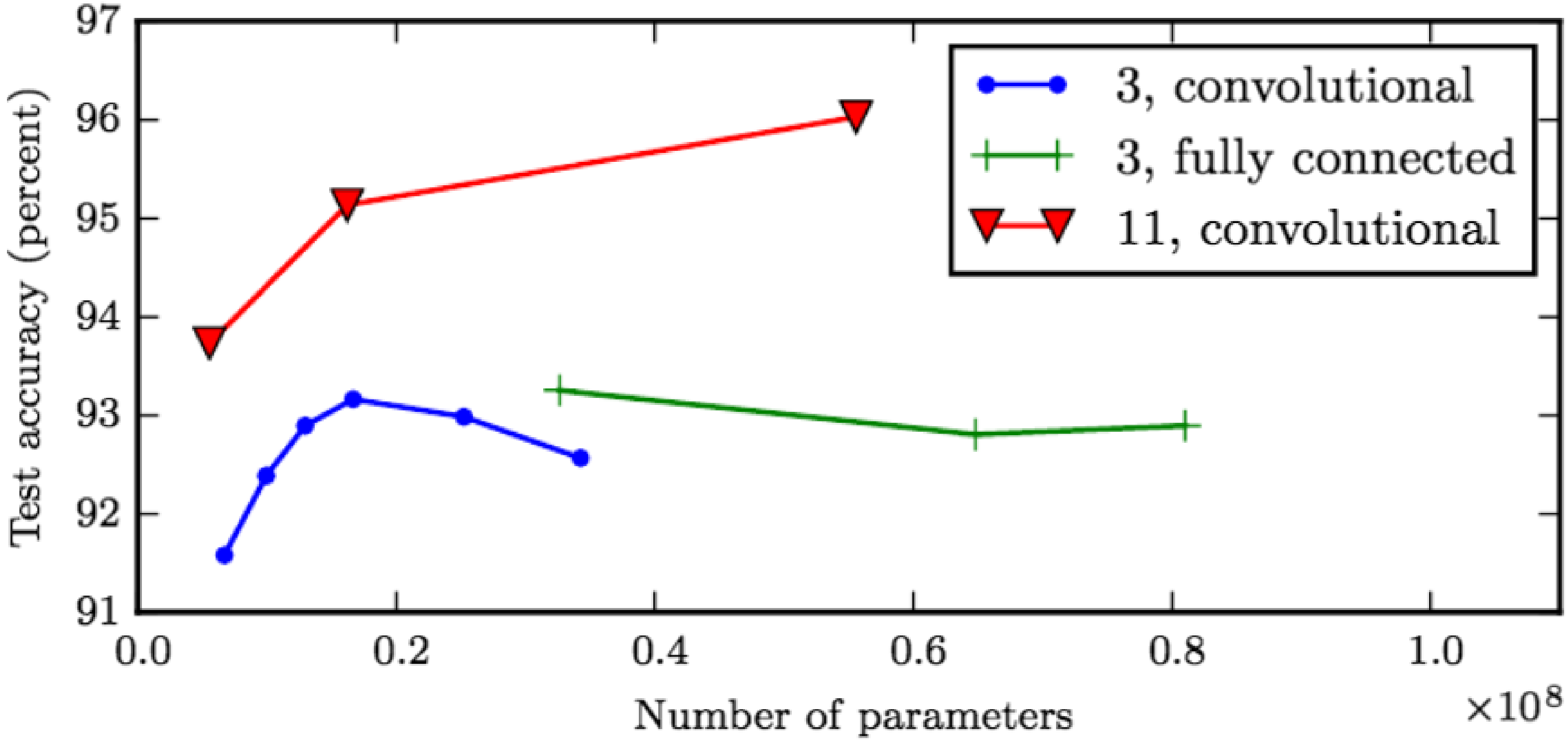


Figure 6.6: Empirical results showing that deeper networks generalize better when used to transcribe multi-digit numbers from photographs of addresses. Data from Goodfellow *et al.* (2014d). The test set accuracy consistently increases with increasing depth. See figure 6.7 for a control experiment demonstrating that other increases to the model size do not yield the same effect.

Efficiency of convnets



Activation Functions

- ReLU: $g(x) = \max(0, x)$
- Leaky ReLU: $g(x) = \max(0, x) + \alpha \min(0, x) \quad (\alpha \approx .01)$
- Tanh: $g(x) = 2\sigma(2x) - 1$
- Radial Basis Functions: $g(x) = \exp(-(w - x)^2/\sigma^2)$
- Softplus: $g(x) = \log(1 + e^x)$
- Hard Tanh: $g(x) = \max(-1, \min(1, x))$
- Maxout: $g(x) = \max_{j \in \mathbb{G}} x_j$
- ….

Architectures

- Some commonly referred to architectures:
  - AlexNet
  - VGG16/19
  - GoogleNet
  - ResNet
  - WideResNet
  - Inception
  - …

Architecture Design and Training Issues

- How many layers? How many hidden units per layer? How to connect layers together? How to optimize?
  - Cost functions
  - L2/L1 regularization
  - Data Set Augmentation
  - Early Stopping
  - Dropout
  - Minibatch Training
  - Momentum
  - Initialization
  - Batch Normalization

Cost Functions

- For regression problems, quadratic error is typical
- For classification, quadratic loss is not as effective
  - Instead one typically uses softmax outputs with cross-entropy error function
  - Discussed earlier, won’t review in depth

$$
\begin{aligned}
J(\theta) &= \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\
&= -\frac{1}{m} [\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]
\end{aligned}
$$

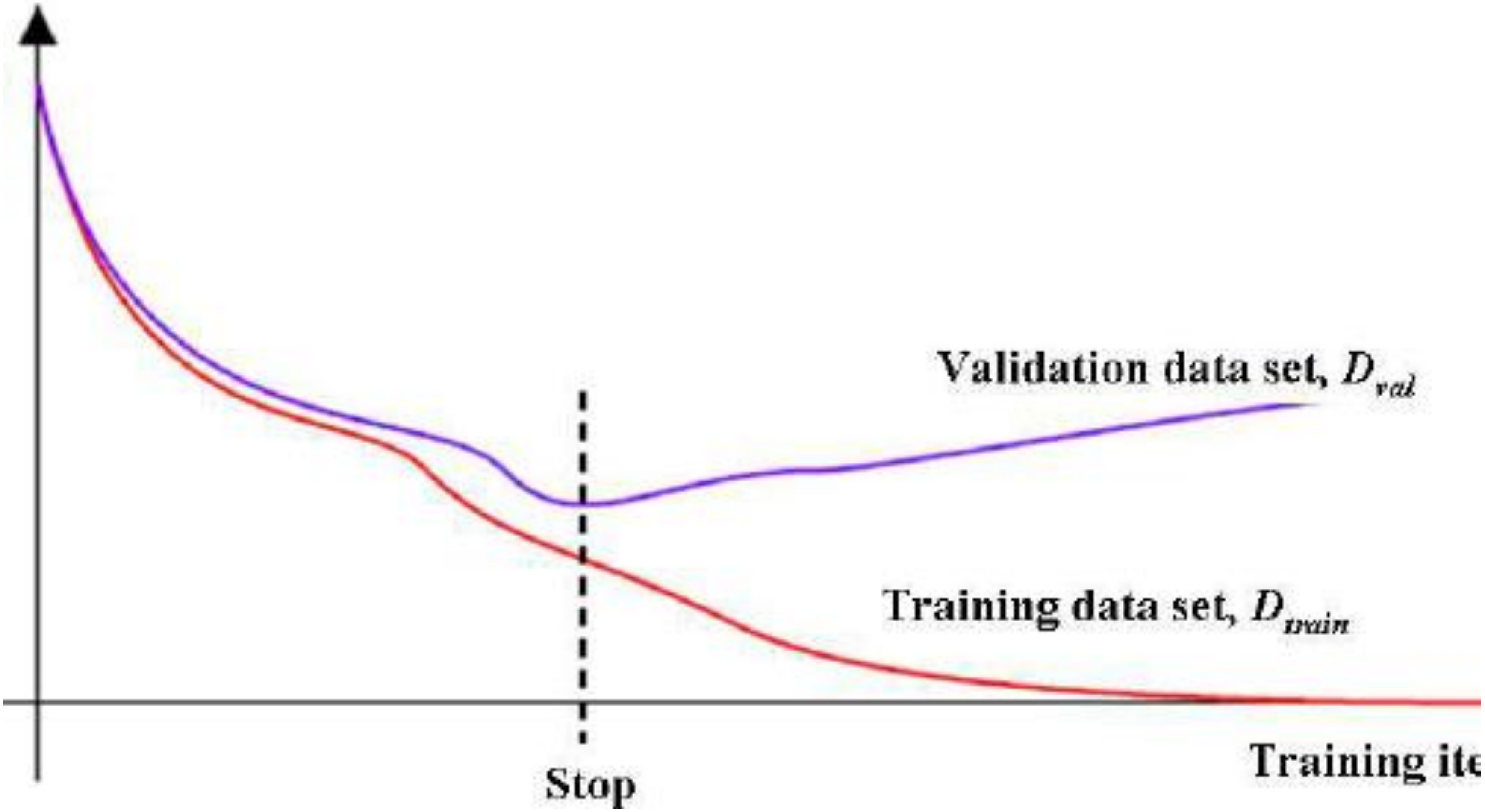
Regularization

- In machine learning, we care about *generalization performance*, not just training error
- With many parameters, models are prone to *overfitting*
- How to regularize?
  - Restrictions on parameter values or function classes
  - Adding terms to the objective function
  - Examples: L2 or L1 regularization

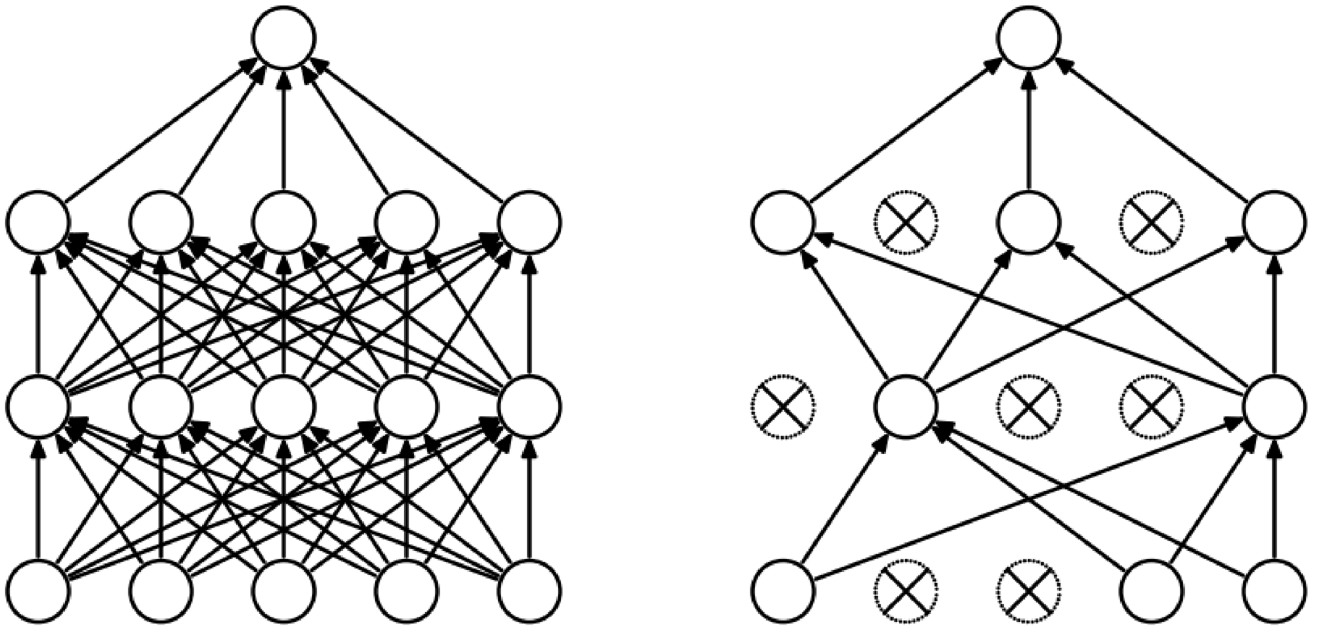
Data Set Augmentation

- The more data, the better for generalization (usually)
- Sometimes we can augment our existing data set
  - Example: for image classification, mirror-image all images to double the size of the training set
  - Injecting noise to training data is also a form of data augmentation

Early Stopping



Dropout



(a) Standard Neural Net

(b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Dropout

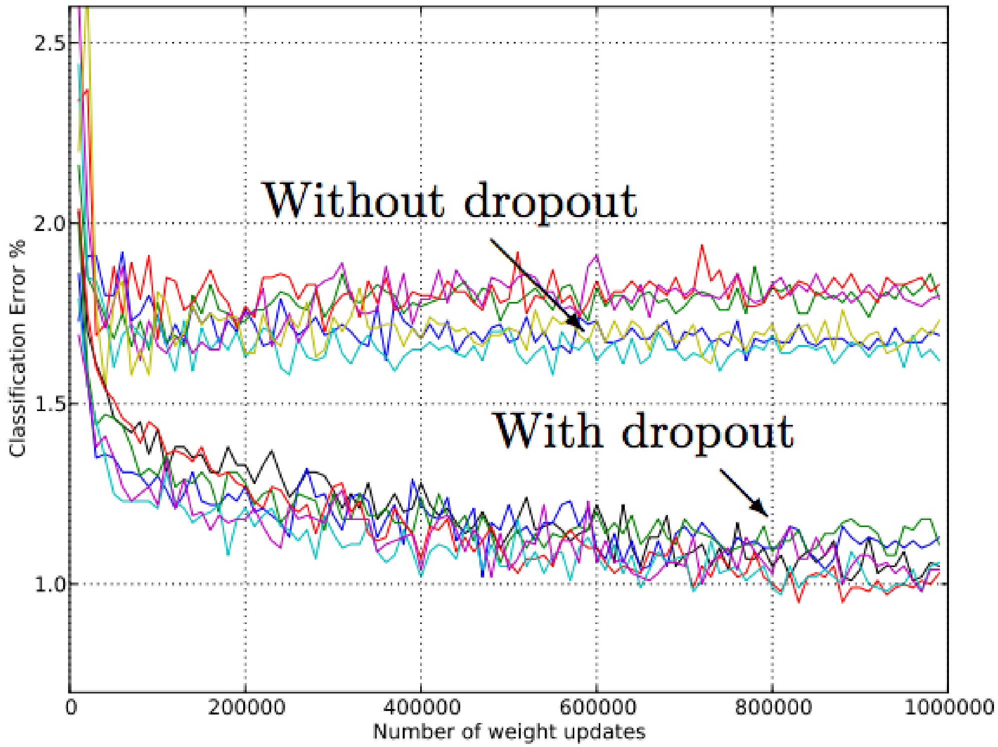


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

Architecture Design and Training Issues

- How many layers? How many hidden units per layer? How to connect layers together? How to optimize?
  - Cost functions
  - *L2/L1 regularization*
  - *Data Set Augmentation*
  - *Early Stopping*
  - *Dropout*
  - Minibatch Training
  - Momentum
  - Initialization
  - Batch Normalization

Avoid Overfitting

Minibatch Training

- Gradient descent uses all training points, fully online (stochastic) methods update using a single point
- Many deep learning methods fall in between
- Example: computing the mean of a set of samples
  - Standard error based on a sample of n points is $\sigma / \sqrt{n}$
  - Consider using 100 versus 10,000 samples
  - Latter requires 100x more computation but reduces error by factor of 10

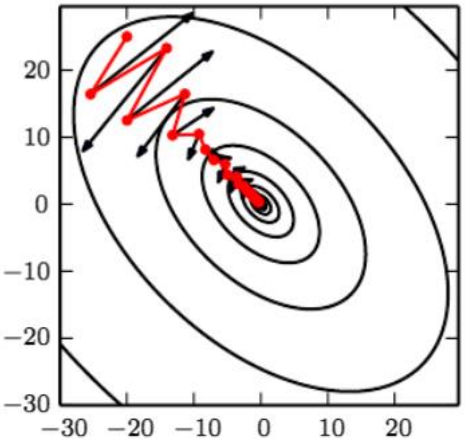
Minibatch Training

- Larger batches provide a more accurate estimate of the gradientIf all examples in the batch are processed in parallel, amount of memory scales with batch size
- Small batches can offer a regularizing effect due to the noise added during the learning process
- When using GPUs it is common for power of 2 batch sizes to offer better runtime; typical sizes range from 32 to 256, with 16 being common for large models
- Minibatches should be selected randomly!

Momentum

- Accumulates an exponentially decaying moving average of past gradients and continues to move in their direction

- exponentially weighed averages can provide us a better estimate which is closer to the actual derivate than our noisy batch calculations



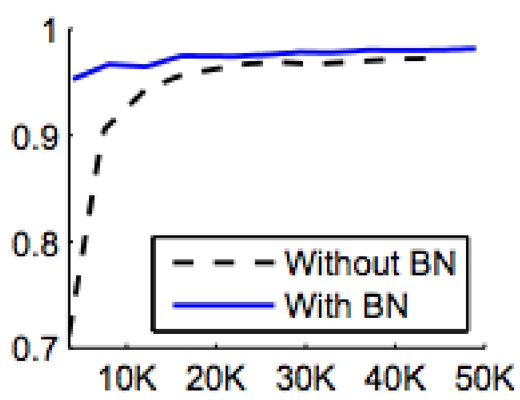
Initialization

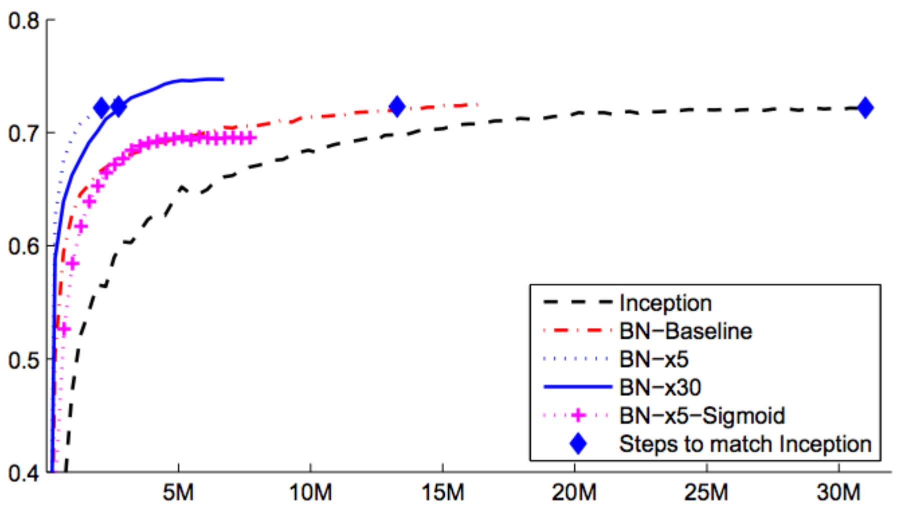
- Important: need to "break symmetry"
  - E.g. Choose weights randomly
- Combined with early stopping, can think of initialization as a prior on the weights
- Usually use uniform or Gaussian weights with a zero-mean
- Examples with m inputs and n outputs:

$$W_{ij} \sim U\left(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\right) \qquad W_{ij} \sim U\left(-\frac{6}{\sqrt{m+n}}, \frac{6}{\sqrt{m+n}}\right)$$

Batch Normalization

- High-level idea: whitening the data at each layer makes training faster
- Left: MNIST, Right: ImageNet





Batch Normalization

Model	Resolution	Crops	Models	Top-1 error	Top-5 error
GoogLeNet ensemble	224	144	7	-	6.67%
Deep Image low-res	256	-	1	-	7.96%
Deep Image high-res	512	-	1	24.88	7.42%
Deep Image ensemble	variable	-	-	-	5.98%
BN-Inception single crop	224	1	1	25.2%	7.82%
BN-Inception multicrop	224	144	1	21.99%	5.82%
BN-Inception ensemble	224	144	6	20.1%	**4.9%***

Figure 4: *Batch-Normalized Inception comparison with previous state of the art on the provided validation set comprising 50000 images. *BN-Inception ensemble has reached 4.82% top-5 error on the 100000 images of the test set of the ImageNet as reported by the test server.*

Data Independence

- NN models converging to the correct solution depends on the iid assumption
- i.e. that our data are independent and identically distributed
- Important to randomly shuffle examples! Otherwise net can fail to converge

Enjoy your Spring Break! ☺