24-09–2024

Class project update

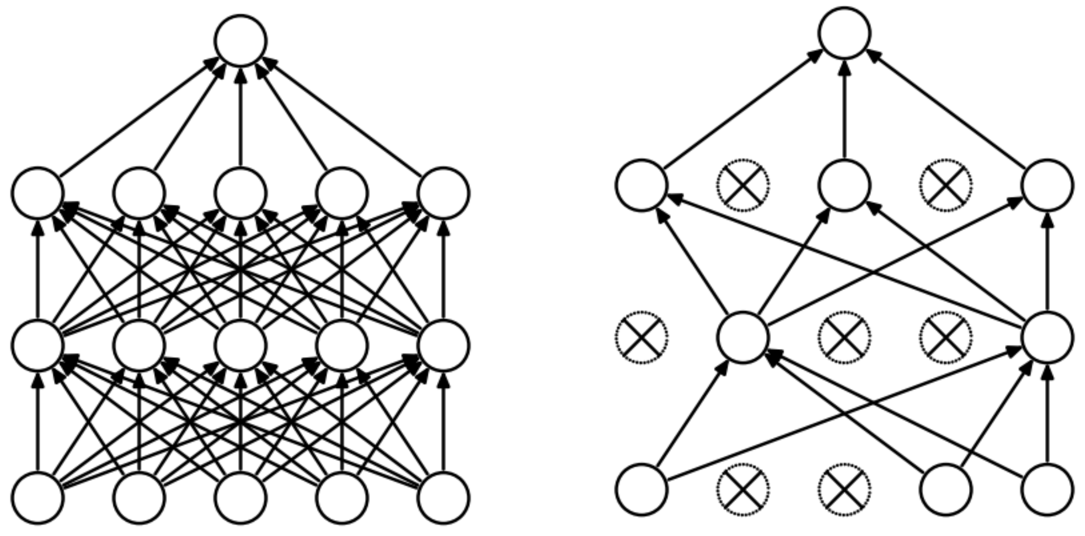
- When?
  - How about 05-10-2024
    - 11 AM to 2:30PM

How to train a CNN ?

- What is different from our MLP training?
- Let’s consider the operations in different layers?
  - Convolution
  - ReLU
  - Pooling
  - Fully connected layers

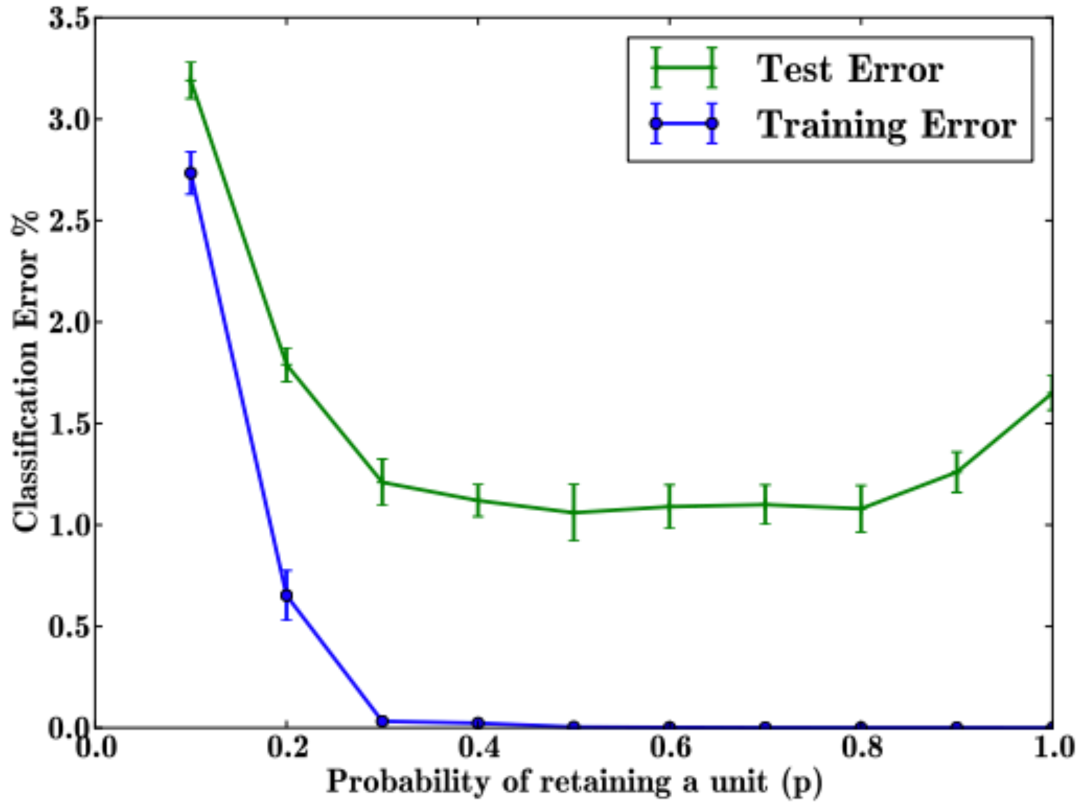
Regularisation

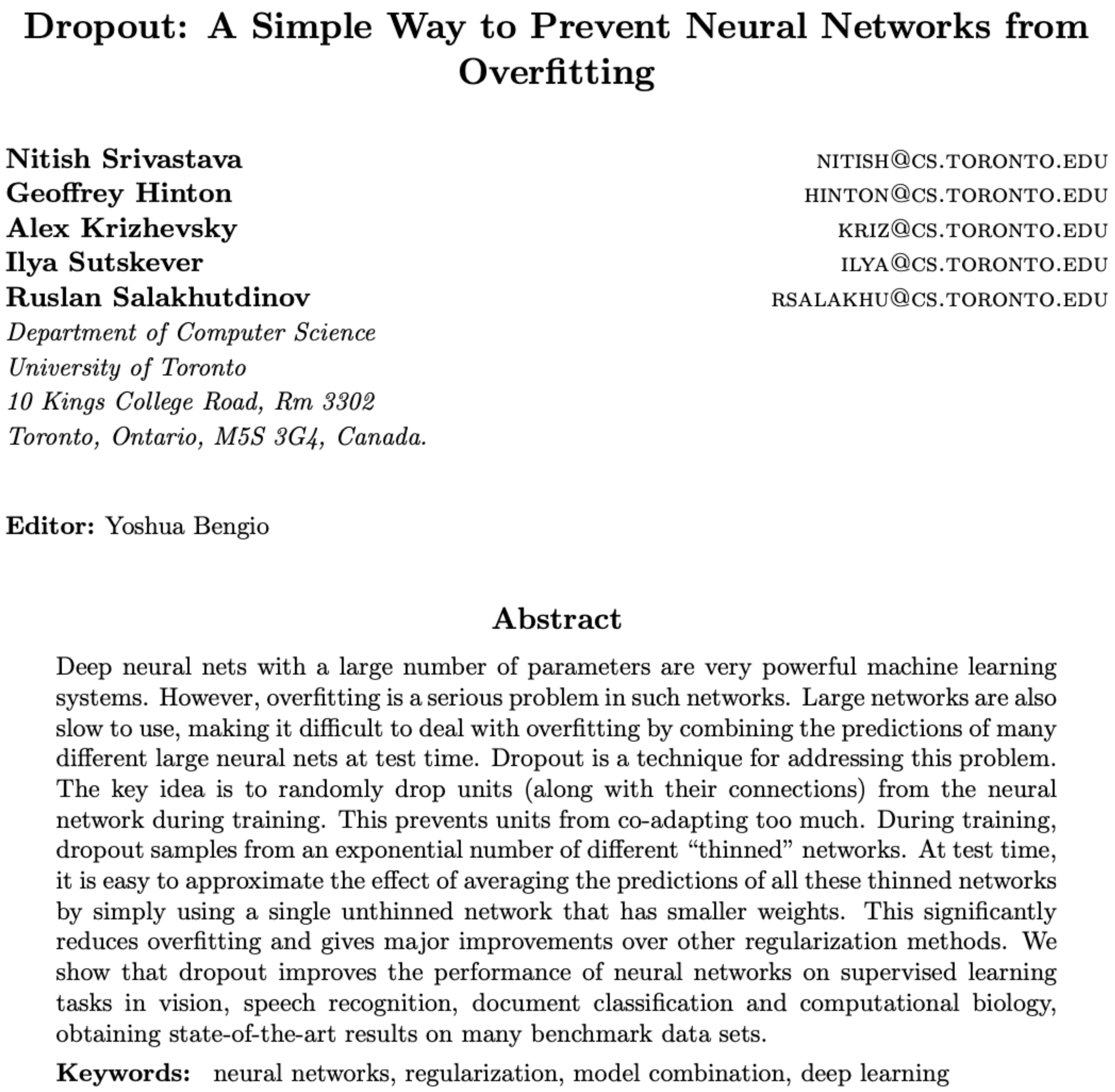
- Think about linear regression and overfitting?
  - L-1, L-2, …. on parameters
- For NN, you can use similar regularizer
  - L-1, L-2, …. on parameters
- Another way to tackle overfitting
  - Dropout
    - $Z_i^{l+1} = W_i^{l+1} Y^l + b_i^{l+1}$
    - $Y_i^{l+1} = f(Z_i^{l+1})$
    - $r_i^l \sim \text{Bernoulli}(p)$
    - $Y^l = r^l * Y^l$



Standard NN

After dropout





Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

Editor: Yoshua Bengio

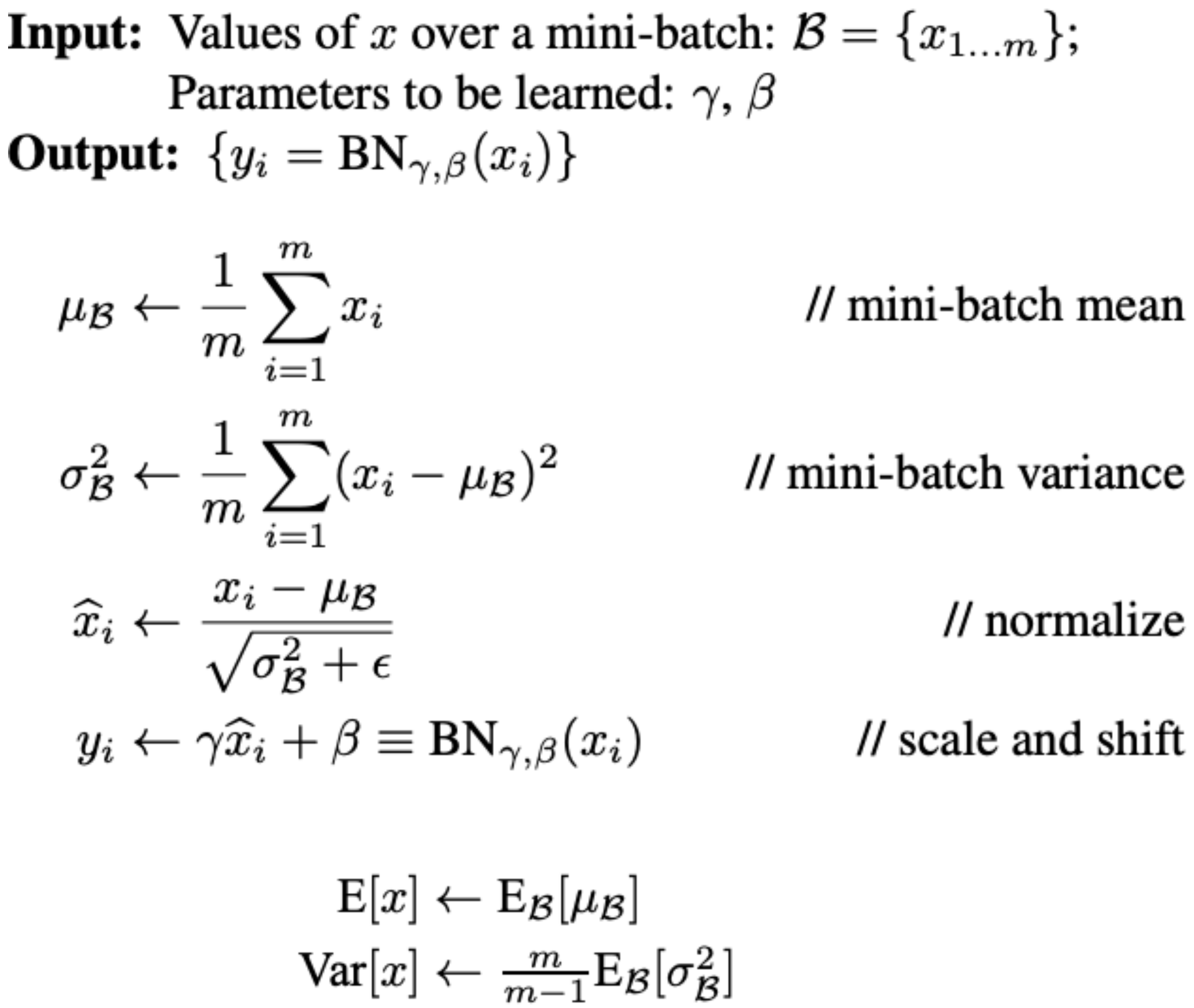
Abstract

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods. We show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets.

Keywords: neural networks, regularization, model combination, deep learning

How can we train a NN faster?

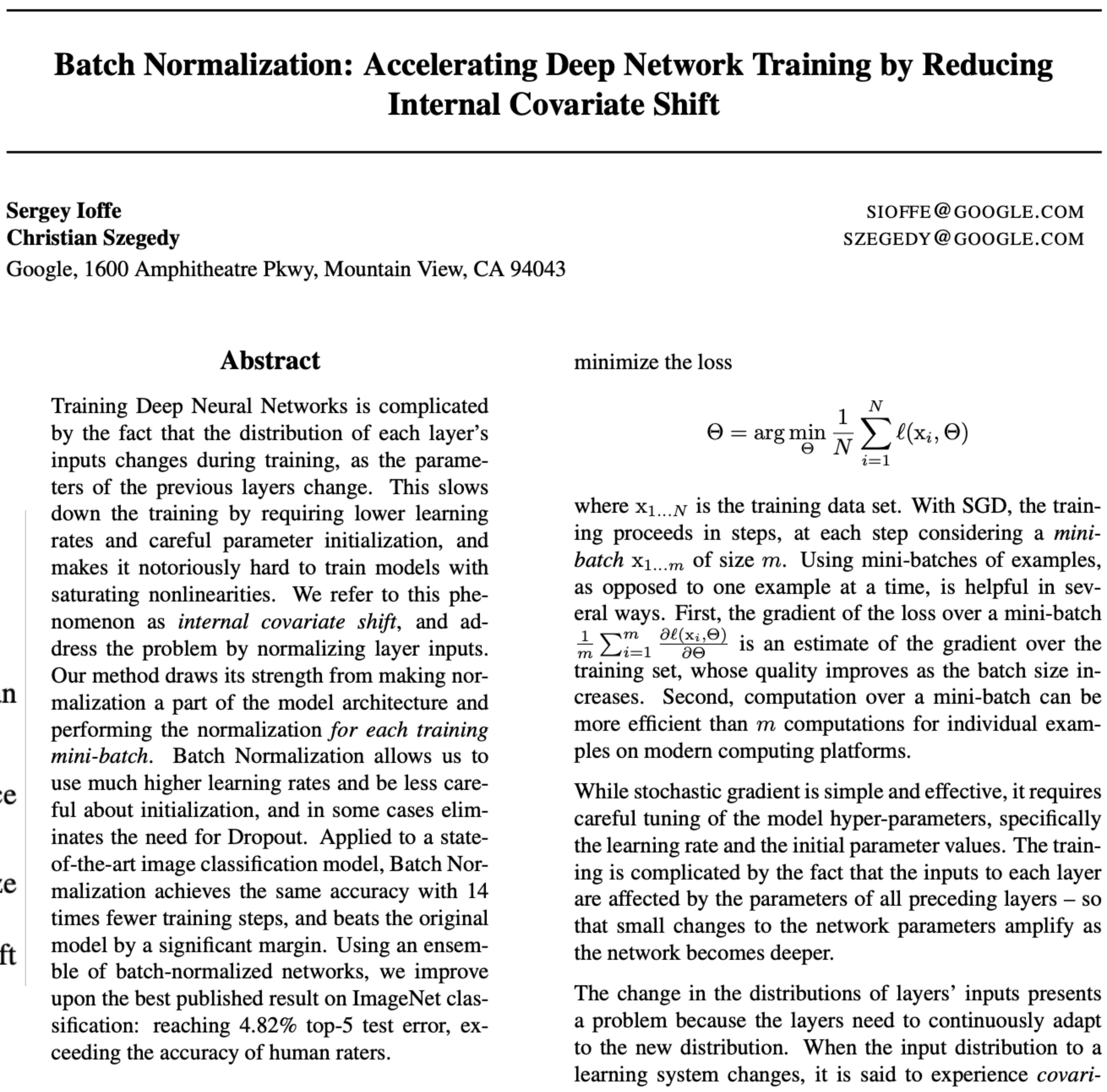
- Good initialisation?
- Learning rate?
- Update rules?
- Any other options?
  - Batch Normalization
  - Normalize at each layer
  - 14 time faster



```
Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
       Parameters to be learned: $\gamma$, $\beta$
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$                        // mini-batch mean
$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\mathcal{B})^2$          // mini-batch variance
$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$                   // normalize
$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$           // scale and shift
```

$$\mathrm{E}[x] \leftarrow \mathrm{E}_\mathcal{B}[\mu_\mathcal{B}]$$
$$\mathrm{Var}[x] \leftarrow \frac{m}{m-1}\mathrm{E}_\mathcal{B}[\sigma_\mathcal{B}^2]$$



Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe SIOFFE@GOOGLE.COM
Christian Szegedy SZEGEDY@GOOGLE.COM
Google, 1600 Amphitheatre Pkwy, Mountain View, CA 94043

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch Normalization allows us to use much higher learning rates and be less careful about initialization, and in some cases eliminates the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.82% top-5 test error, exceeding the accuracy of human raters.

minimize the loss

$$\Theta = \arg\min_\Theta \frac{1}{N}\sum_{i=1}^{N}\ell(\mathrm{x}_i, \Theta)$$

where $\mathrm{x}_{1...N}$ is the training data set. With SGD, the training proceeds in steps, at each step considering a *mini-batch* $\mathrm{x}_{1...m}$ of size m. Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch $\frac{1}{m}\sum_{i=1}^{m}\frac{\partial\ell(\mathrm{x}_i,\Theta)}{\partial\Theta}$ is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a mini-batch can be more efficient than m computations for individual examples on modern computing platforms.

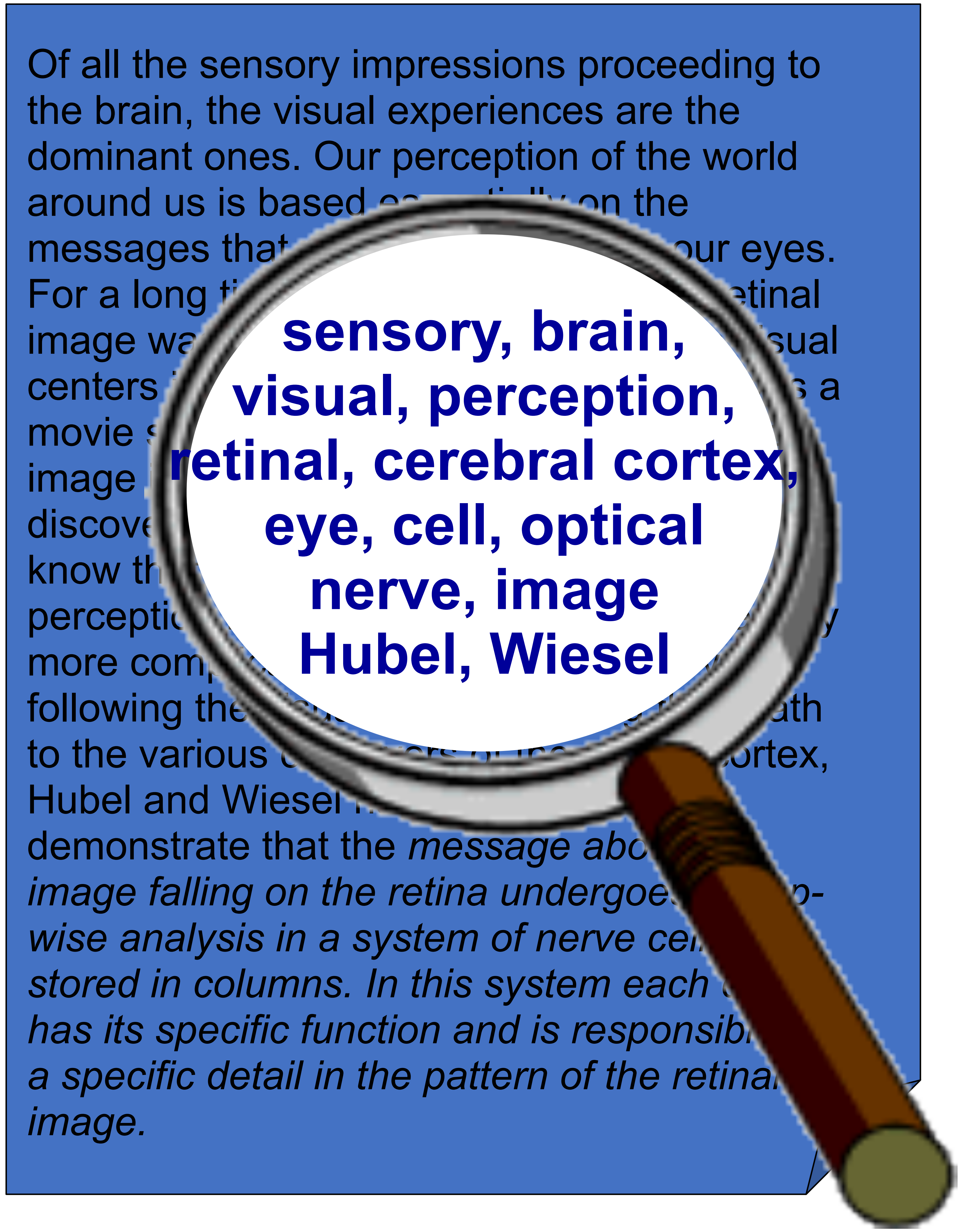
While stochastic gradient is simple and effective, it requires careful tuning of the model hyper-parameters, specifically the learning rate and the initial parameter values. The training is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers – so that small changes to the network parameters amplify as the network becomes deeper.

The change in the distributions of layers' inputs presents a problem because the layers need to continuously adapt to the new distribution. When the input distribution to a learning system changes, it is said to experience *covari-*

Application of DL in Natural Language Processing (NLP)

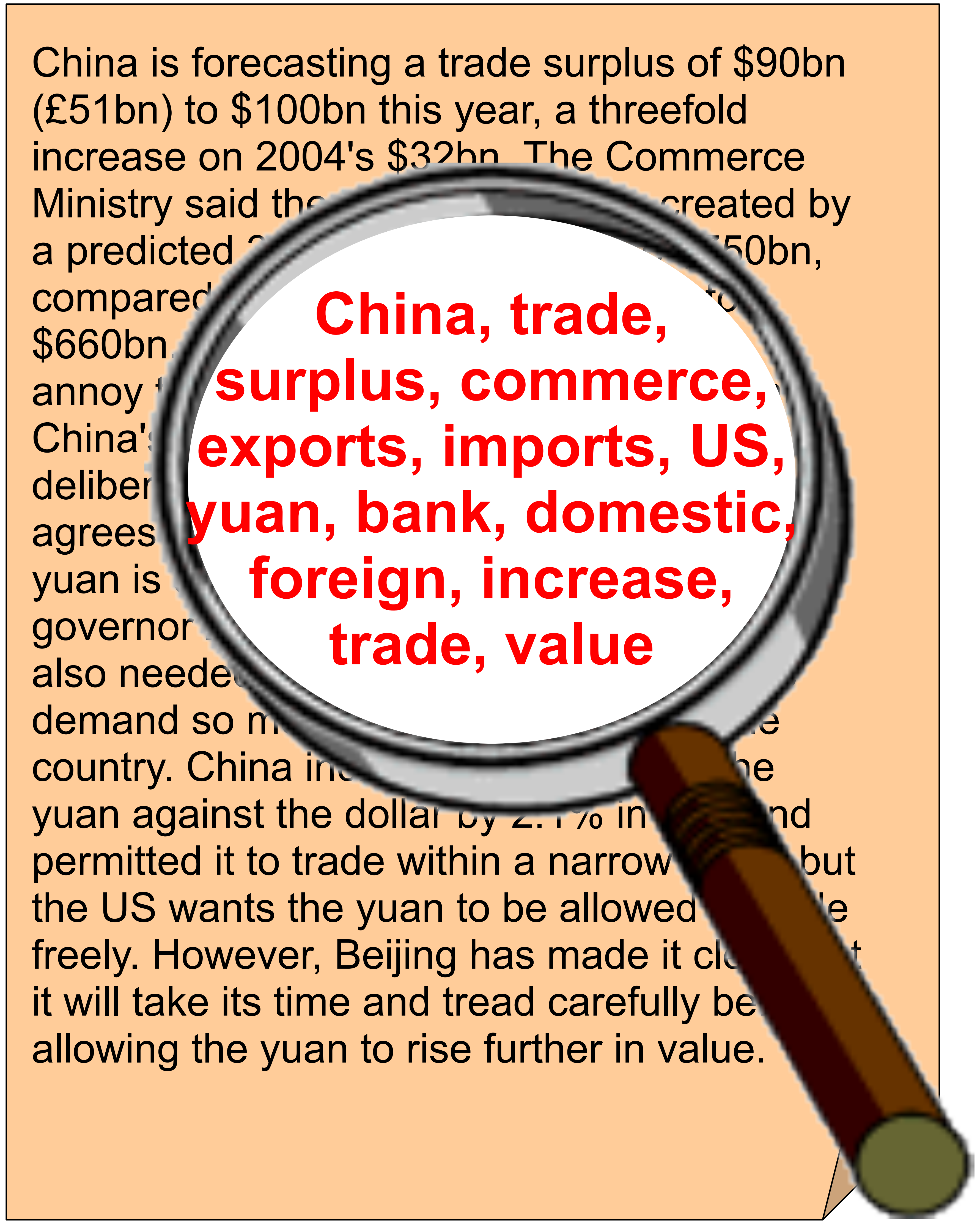
Language model
(Classical models)

Analogy to documents



Of all the sensory impressions proceeding to
the brain, the visual experiences are the
dominant ones. Our perception of the world
around us is based on the
messages that our eyes.
For a long t etinal
image wa sual
centers a
movie
image
discove
know th
percepti
more com
following the ath
to the various ortex,
Hubel and Wiesel
demonstrate that the *message abo*
image falling on the retina undergoes p-
wise analysis in a system of nerve cel
stored in columns. In this system each
has its specific function and is responsib
a specific detail in the pattern of the retinal
image.

sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel



China is forecasting a trade surplus of $90bn
(£51bn) to $100bn this year, a threefold
increase on 2004's $32bn. The Commerce
Ministry said the created by
a predicted 50bn,
compared
$660bn
annoy
China's
deliber
agrees
yuan is
governor
also neede
demand so m e
country. China in he
yuan against the dollar by 2.1% in nd
permitted it to trade within a narrow but
the US wants the yuan to be allowed e
freely. However, Beijing has made it cl t
it will take its time and tread carefully be
allowing the yuan to rise further in value.

China, trade,
surplus, commerce,
exports, imports, US,
yuan, bank, domestic,
foreign, increase,
trade, value

Bag-of-Words model

- Text normalization?
- How many words?
- Binary or count?

Language model

- Example
  - All RKMVERI students are XXX
    - Possibilities: Excellent/good/average …
    - Not likely: Table/refrigerator/…
  - Sentence score:
    - Sentence-1: “It is a pity that the existing system of education did not enable a person to stand on his own feet, nor did it teach him self-confidence and self-respect”
    - Sentence-2: Pity is it the system existing stand of system not person education did to on nor, own his did feet teach it self-confidence him self-respect and
    - What about $p(Sentence-1)$ and $p(Sentence-2)$?

Language model: N-gram

- $p(\text{the} \mid \text{It is a pity that})$?
- $$p(\text{the} \mid \text{It is a pity that}) = \frac{Count(\text{It is a pity that the})}{\sum_x Count(\text{It is a pity that x})}$$
  $$= \frac{Count(\text{It is a pity that the})}{Count(\text{It is a pity that})}$$
- $p(\text{It is a pity} \cdots \text{and self-respect.}) = p(\text{It})p(\text{is|It})p(\text{a} \mid \text{It is}) \cdots p(\text{self-repect} \mid \text{It is} \cdots \text{and})$
- Assumption: Markov
  - $p(\text{It is a pity} \cdots \text{and self-respect.}) = p(\text{It})p(\text{is|It})p(\text{a} \mid \text{It}) \cdots p(\text{self-repect} \mid \text{and})$

Jurafsky & Martin “Speech and Language Processing, 3rd ed., 2023

N-gram: toy example

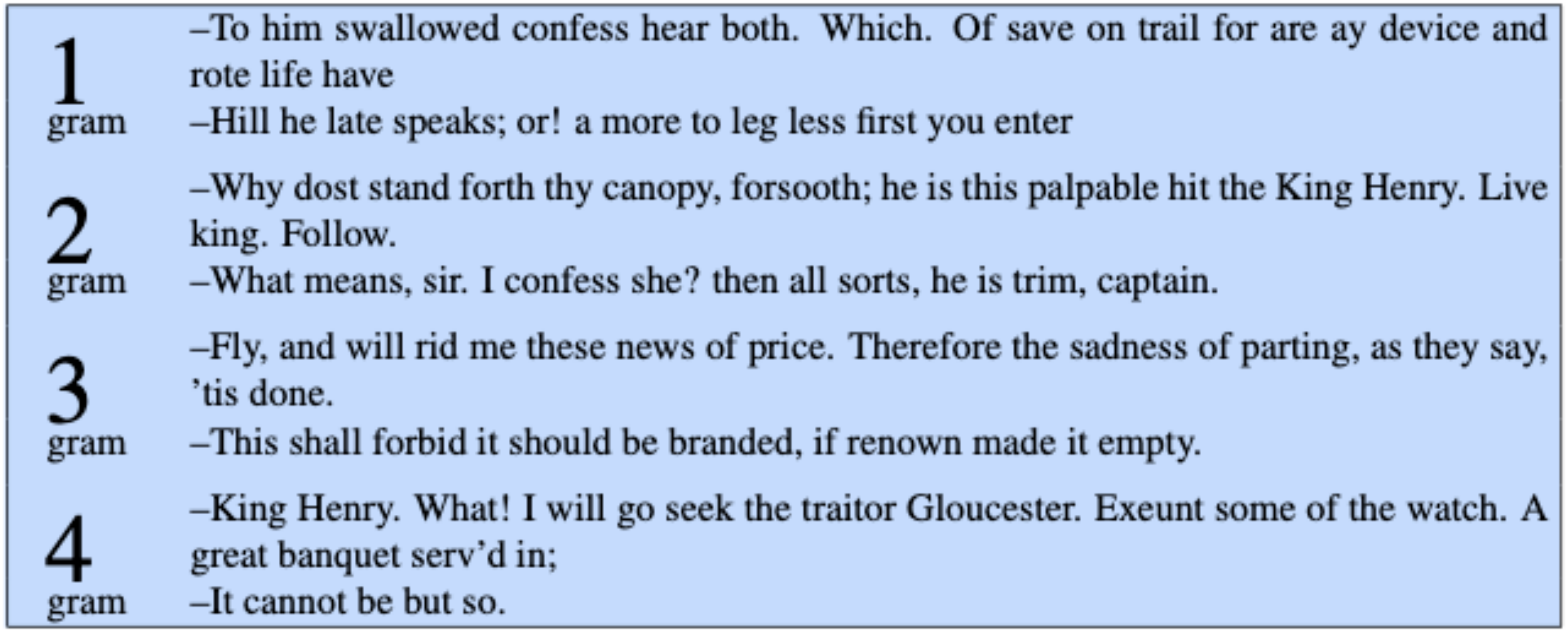
- Consider a toy example (corpus):
  - <s> I am Amal </s>
  - <s> I am studying in the RKMVERI </s>
  - <s> I stayed at the hostel </s>
  - <s> Why am I attending the NLP course? </s>
  - <s> Am I studying enough for my class project?</s>
- Calculate some 2-gram/bigram probabilities for the above corpus
  - $p(\text{I} \mid \text{<s>}) = \frac{3}{5}$
  - $p(\text{Why} \mid \text{<s>}) = \frac{1}{5}$
  - $p(\text{am} \mid \text{I}) = \frac{1}{5}$
  - $p(\text{enough} \mid \text{studying}) = \frac{1}{2}$
  - …

N-gram: practical issues

- What happen when N is large ?
  - Say 5, 6, …
  - $p(\text{I} \mid \text{<s><s><s><s>})$
- Numerical stability
  - Take log

Sampling from a language model

- Generate new sentences?
- How?
  - Define a probability line based on your model
  - Generate a random number between 0 and 1
  - Find the point on the probability line and pick the word corresponding to that point



1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

Jurafsky & Martin “Speech and Language Processing, 3rd ed., 2023

N-gram: issues

- Practical issues
  - What happen when N is large ?
    - Say 5, 6, …
    - $p(\text{I} \mid \text{<s><s><s><s>})$
  - Numerical stability
    - Take log
- Other issues
  - Unknown words (out of vocabulary)
    - Use special token <UNK>?
  - Some N-gram counts might be zero but they might appear in the test case?
    - Smoothing

Smoothing

- Laplace smoothing
  - ‣ +1 to all the N-gram counts
  - ‣ $p(w_i) = \frac{c_i}{C}$ -> $p_{Laplace}(w_i) = \frac{c_i + 1}{C + |V|}$
  - ‣ Discount
  - ‣ Can you see any problem here?
- Add-k smoothing
  - ‣ $p(w_i) = \frac{c_i}{C}$->$p_{Laplace}(w_i) = \frac{c_i + 1}{C + |V|}$->$p_{Add-k}(w_i) = \frac{c_i + k}{C + k|V|}$
    - $k = 0.5, 0.05, 0.005, \cdots$

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Smoothing (cont.)

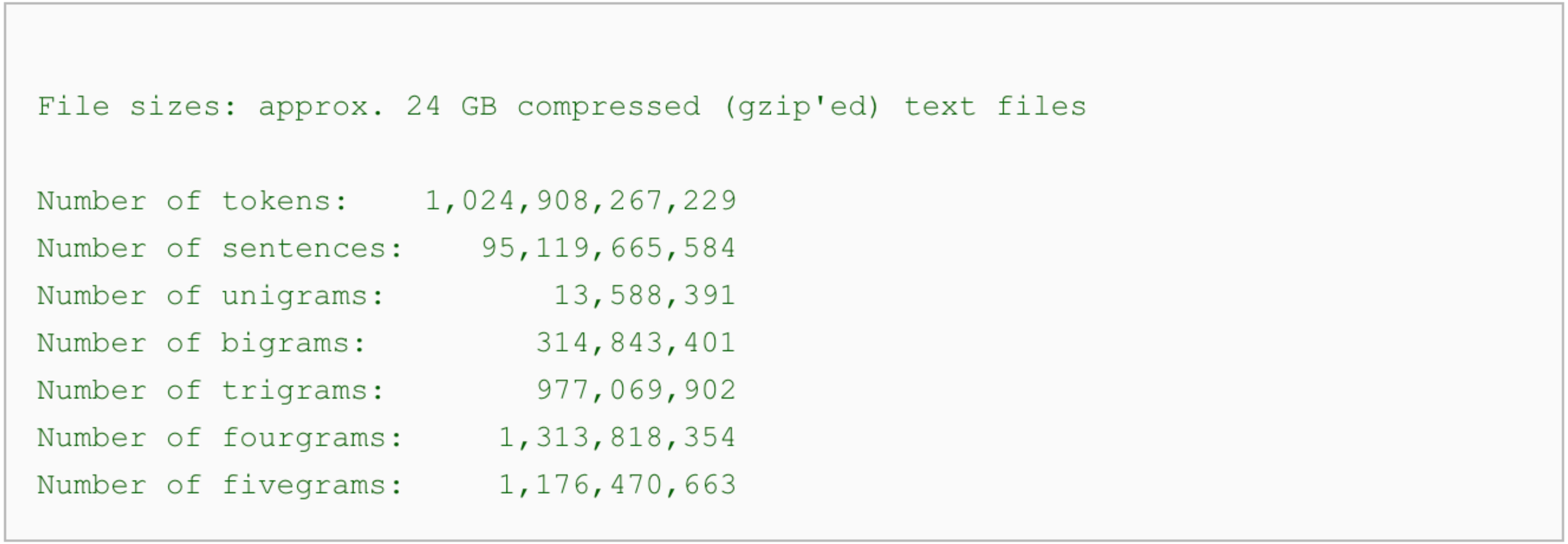
- Backoff
  - If N-gram interested has zero count then we can approximate it by backing off to the (N-1)-gram
  - Continue backing off until we found a non-zero count in our history (Ex. (N-2), (N-3),…,1-gram)
- Interpolation
  - $\hat{p}(w_n | w_{n-2}w_{n-1}) = \lambda_1 p(w_n) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n | w_{n-2}w_{n-1}); \sum_i \lambda_i = 1$

    - $\lambda_1 \rightarrow \lambda_1(w_{n-2}w_{n-1}); \lambda_2 \rightarrow \lambda_2(w_{n-2}w_{n-1}); \lambda_3 \rightarrow \lambda_3(w_{n-2}w_{n-1})$
- Katz's back-off
  - $p_{BO}(w_n | w_{n-N+1:n-1}) = \begin{cases} p^*(w_n | w_{n-N+1:n-1}) & \text{if } C(w_{n-N+1:n}) > 0 \\ \alpha(w_{n-N+1:n-1}) p_{BO}(w_n | w_{n-N+2:n-1}) & \text{otherwise} \end{cases}$

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Smoothing (cont.)

- Kneser-ney shooting, 1995
- Google n-gram model (2006):
  - https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html



```
File sizes: approx. 24 GB compressed (gzip'ed) text files

Number of tokens:    1,024,908,267,229
Number of sentences:    95,119,665,584
Number of unigrams:         13,588,391
Number of bigrams:         314,843,401
Number of trigrams:        977,069,902
Number of fourgrams:     1,313,818,354
Number of fivegrams:     1,176,470,663
```

Language model evaluation

- Extrinsic evaluation - application specific
- Intrinsic evaluation - independent of any application
- Perplexity $(W = w_1 w_2 \ldots w_N)$:
  - $PP(W) = p(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$
  - For 2-gram Model
    - $$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{p(w_i \,|\, w_{i-1})}}$$
- Can you compare 2-gram and 3-gram model on a (our) toy corpus?

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023