Recent advances in model compression

Krzysztof J. Geras

NEW YORK UNIVERSITY

Joint work with **Rich Caruana**, **Gregor Urban**, **Abdel-rahman Mohamed**, **Charles Sutton**, **Shengjie Wang**, **Özlem Aslan**, **Samira Ebrahimi Kahou**, **Matthai Philipose** and **Matthew Richardson**

TFML 2017

Neural networks

Neural networks

- They work amazingly well.

Neural networks

- They work amazingly well.
- We are largely limited to empirical exploration.

THE NEURAL NETWORK ZOO

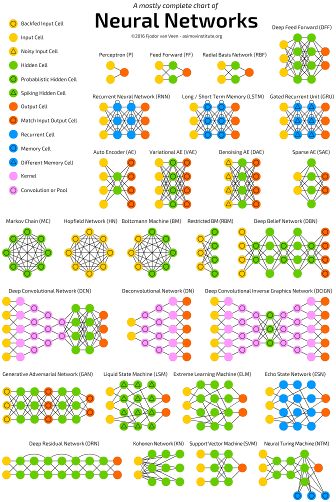
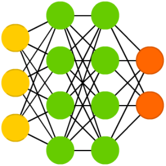
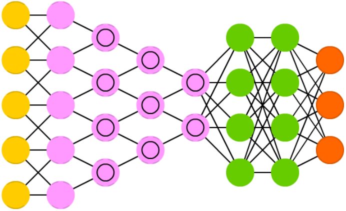


Figure from asimovinstitute.org/neural-network-zoo/.

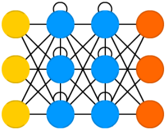
Deep Feed Forward (DFF)



Deep Convolutional Network (DCN)

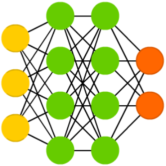


Recurrent Neural Network (RNN)

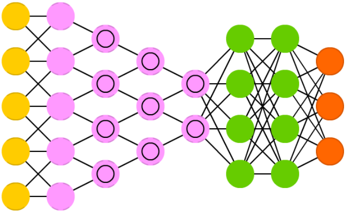


Learnability Representability

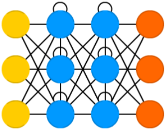
Deep Feed Forward (DFF)



Deep Convolutional Network (DCN)



Recurrent Neural Network (RNN)



Learnability $\neq$ Representability

Model compression (aka Knowledge Distillation)

- Idea: take predictions from a big, complex, accurate classifier (a *teacher*) and train a simpler model (a *student*) using them instead of training labels.

Model compression (aka Knowledge Distillation)

- Idea: take predictions from a big, complex, accurate classifier (a *teacher*) and train a simpler model (a *student*) using them instead of training labels.
- That is, optimise

$$L = -\sum_j \sum_c p(c|\mathbf{x}_j) \log q(c|\mathbf{x}_j),$$

where $p(c|\mathbf{x}_j)$ is teacher's posterior probability of class c given $\mathbf{x}_j$ and $q(c|\mathbf{x}_j)$ is the same for the student.

Model compression (aka Knowledge Distillation)

- Alternatively,

$$L = \lambda \left[-\sum_j \sum_c p(c|\mathbf{x}_j) \log q(c|\mathbf{x}_j) \right] + (1-\lambda) \left[-\sum_j \log q(y_j|\mathbf{x}_j) \right],$$

where $p(c|\mathbf{x}_j)$ is teacher's posterior probability of class c given $\mathbf{x}_j$ and $q(c|\mathbf{x}_j)$ is the same for the student.

Model compression

Why does that work?

Model compression

Why does that work?

Hypotheses:

Model compression

Why does that work?

Hypotheses:

- Each example shown to the student model is given with a richer supervision signal.

Model compression

Why does that work?

Hypotheses:

- Each example shown to the student model is given with a richer supervision signal.
- Cleans noisy labels.

Model compression

Why does that work?

Hypotheses:

- Each example shown to the student model is given with a richer supervision signal.
- Cleans noisy labels.
- A way to transfer an inductive bias between models.

Model compression

- Large ensemble $\rightarrow$ single non-convolutional net (Bucila et al., 2006).

Model compression

- Large ensemble $\rightarrow$ single non-convolutional net (Bucila et al., 2006).
- Ensemble of deep convolutional nets $\rightarrow$ single shallow non-convolutional net (Ba and Caruana, 2014).

DO DEEP NETS REALLY NEED TO BE DEEP?

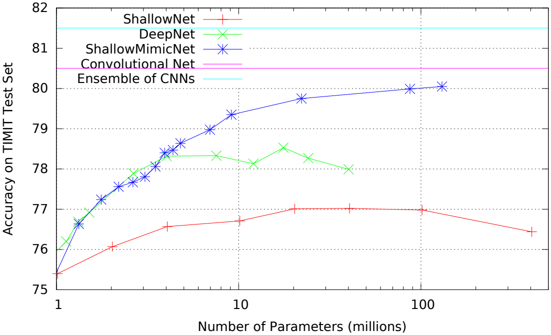


Figure from Ba and Caruana (2014).

Model compression

- Large ensemble $\rightarrow$ single non-convolutional net (Bucila et al., 2006).
- Ensemble of deep convolutional nets $\rightarrow$ single shallow non-convolutional net (Ba and Caruana, 2014).
- Ensemble of deep non-convolutional nets $\rightarrow$ single deep non-convolutional net (Hinton et al., 2014).

Model compression

- Large ensemble $\rightarrow$ single non-convolutional net (Bucila et al., 2006).
- Ensemble of deep convolutional nets $\rightarrow$ single shallow non-convolutional net (Ba and Caruana, 2014).
- Ensemble of deep non-convolutional nets $\rightarrow$ single deep non-convolutional net (Hinton et al., 2014).
- Ensemble of very deep convolutional nets $\rightarrow$ single shallow convolutional net (Urban et al., 2016).

Model compression

- Large ensemble $\rightarrow$ single non-convolutional net (Bucila et al., 2006).
- Ensemble of deep convolutional nets $\rightarrow$ single shallow non-convolutional net (Ba and Caruana, 2014).
- Ensemble of deep non-convolutional nets $\rightarrow$ single deep non-convolutional net (Hinton et al., 2014).
- Ensemble of very deep convolutional nets $\rightarrow$ single shallow convolutional net (Urban et al., 2016).
- Ensemble of deep recurrent nets $\rightarrow$ single deep convolutional net (Geras et al., 2016).

Do deep convolutional nets really need to be deep (or even convolutional)?

- We know that fully connected nets are compressible.

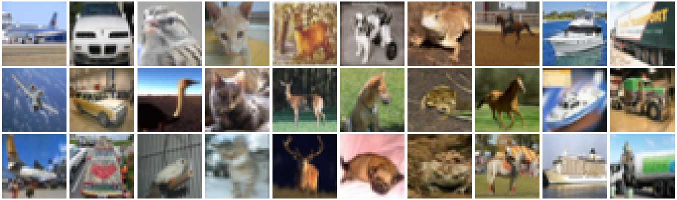
Do deep convolutional nets really need to be deep (or even convolutional)?

- We know that fully connected nets are compressible.
- **Question 1.** Can we compress deep convolutional networks into shallow convolutional networks?

Do deep convolutional nets really need to be deep (or even convolutional)?

- We know that fully connected nets are compressible.
- **Question 1.** Can we compress deep convolutional networks into shallow convolutional networks?
- **Question 2.** Can we compress deep convolutional networks into fully connected networks?

CIFAR-10 DATA SET



- Labelled subset of the Tiny 80M images data set.
- 60k 32x32 RGB images.
- 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, sheep, truck.
- Each class contains 6k images.

Training the teacher and the students

- The teacher: 8 convolutional layers and 2 fully connected layers.

Training the teacher and the students

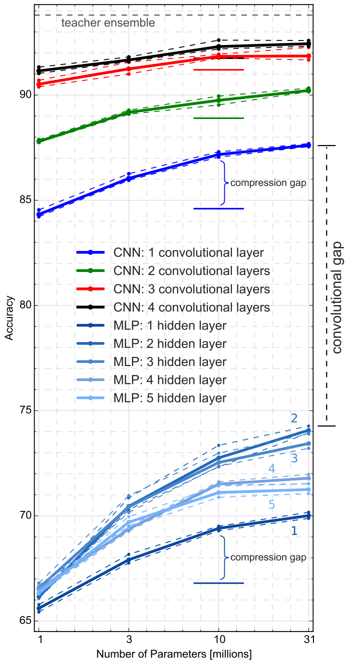
- The teacher: 8 convolutional layers and 2 fully connected layers.
- Various possible student architectures.

Training the teacher and the students

- The teacher: 8 convolutional layers and 2 fully connected layers.
- Various possible student architectures.
- We need to be extremely careful.

Training the teacher and the students

- The teacher: 8 convolutional layers and 2 fully connected layers.
- Various possible student architectures.
- We need to be extremely careful.
- We use Bayesian optimisation to find the best hyperparameters.



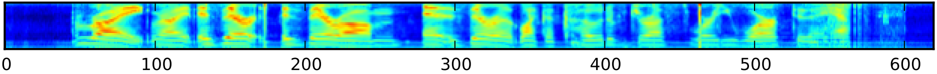
teacher ensemble
compression gap
compression gap
convolutional gap
CNN: 1 convolutional layer
CNN: 2 convolutional layers
CNN: 3 convolutional layers
CNN: 4 convolutional layers
MLP: 1 hidden layer
MLP: 2 hidden layer
MLP: 3 hidden layer
MLP: 4 hidden layer
MLP: 5 hidden layer
Accuracy
Number of Parameters [millions]
1
2
3
4
5
90
85
80
75
70
65
1
3
10
31

Deep convolutional nets really need to be deep.

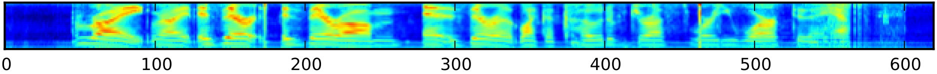
Deep convolutional nets really need to be deep. And convolutional.

Deep convolutional nets really need to be deep. And convolutional. But perhaps not that deep.

Speech recognition $\stackrel{?}{\approx}$ object recognition

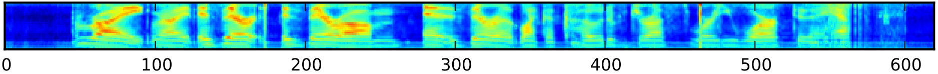


Speech recognition $\stackrel{?}{\approx}$ object recognition



Speech recognition framework:

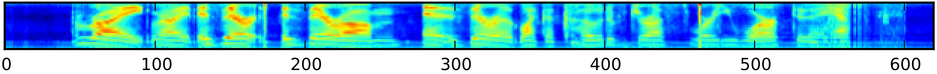
SPEECH RECOGNITION $\overset{?}{\approx}$ OBJECT RECOGNITION



Speech recognition framework:

- Sample many windows of speech. Train a classifier.

Speech recognition $\stackrel{?}{\approx}$ object recognition



Speech recognition framework:

- Sample many windows of speech. Train a classifier.
- Use decoding to get words.

The Switchboard data set

- A benchmark for speech recognition.

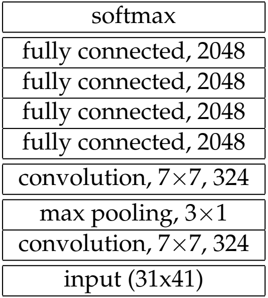
The Switchboard data set

- A benchmark for speech recognition.
- Very large, 309 hours of speech, 18 GB.

The Switchboard data set

- A benchmark for speech recognition.
- Very large, 309 hours of speech, 18 GB.
- We sample training examples of size 31×41, 9000 output classes.

CNNs for speech



CNNs for speech

softmax
fully connected, 2048
fully connected, 2048
fully connected, 2048
fully connected, 2048
convolution, 7×7, 324
max pooling, 3×1
convolution, 7×7, 324
input (31x41)

softmax
fully connected, 4096
fully connected, 4096
max pooling, 2×2
convolution, 3×3, 384
convolution, 3×3, 384
convolution, 3×3, 384
max pooling, 2×2
convolution, 3×3, 192
convolution, 3×3, 192
convolution, 3×3, 192
max pooling, 2×2
convolution, 3×3, 96
convolution, 3×3, 96
input (31x41)

CNNs for speech

softmax
fully connected, 2048
fully connected, 2048
fully connected, 2048
fully connected, 2048
convolution, 7×7, 324
max pooling, 3×1
convolution, 7×7, 324
input (31x41)

Sainath et al.-style CNN

softmax
fully connected, 4096
fully connected, 4096
max pooling, 2×2
convolution, 3×3, 384
convolution, 3×3, 384
convolution, 3×3, 384
max pooling, 2×2
convolution, 3×3, 192
convolution, 3×3, 192
convolution, 3×3, 192
max pooling, 2×2
convolution, 3×3, 96
convolution, 3×3, 96
input (31x41)

vision-style CNN

CNNs vs LSTMs on the Switchboard data set

	frame error rate	word error rate
Sainath et al.-style CNN	37.93%	15.5
vision-style CNN	35.51%	14.1
LSTM	34.15%	14.4

CNNs vs LSTMs on the Switchboard data set

	frame error rate	word error rate
Sainath et al.-style CNN	37.93%	15.5
vision-style CNN	35.51%	14.1
LSTM	34.15%	14.4

CNNs vs LSTMs on the Switchboard data set

	frame error rate	word error rate
Sainath et al.-style CNN	37.93%	15.5
vision-style CNN	35.51%	14.1
LSTM	34.15%	14.4

CNNs vs LSTMs for speech

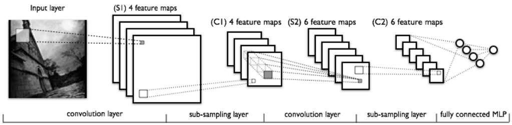
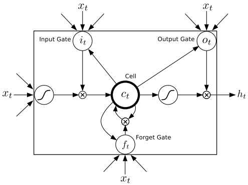


Figure from deeplearning.net



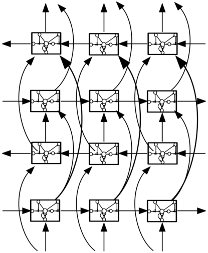


Figure from Graves et al. (2013)

Can we do better?

- Different network structures $\rightarrow$ different inductive biases.

Can we do better?

- Different network structures $\rightarrow$ different inductive biases.
- **Can we have two models in one?**

Can we do better?

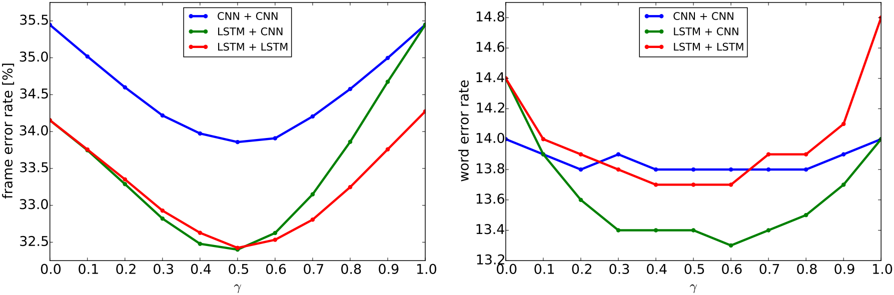
- Different network structures $\rightarrow$ different inductive biases.
- **Can we have two models in one?**
- Yes, there is an easy way to do this - ensembling.

ENSEMBLING

$$p(y|\mathbf{x}_i) = \gamma p_{\text{LSTM}}(y|\mathbf{x}_i) + (1-\gamma) p_{\text{CNN}}(y|\mathbf{x}_i)$$

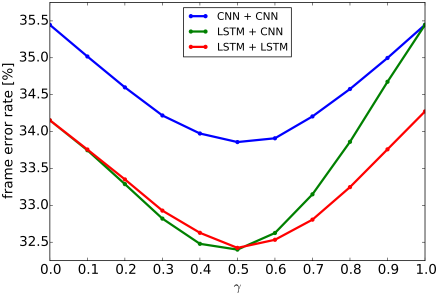
Ensembling

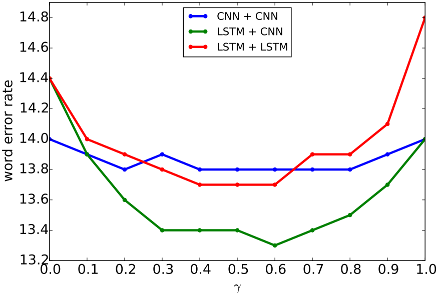
$$p(y|\mathbf{x}_i) = \gamma p_{\mathrm{LSTM}}(y|\mathbf{x}_i) + (1-\gamma)p_{\mathrm{CNN}}(y|\mathbf{x}_i)$$



Ensembling

$$p(y|\mathbf{x}_i) = \gamma p_{\mathrm{LSTM}}(y|\mathbf{x}_i) + (1-\gamma) p_{\mathrm{CNN}}(y|\mathbf{x}_i)$$





Big issue: LSTM is **6** times slower than the CNN. We need to have two models in one CNN.

How to do compression with Switchboard

- Very large data set, 309 hours of speech, 18 GB.

How to do compression with Switchboard

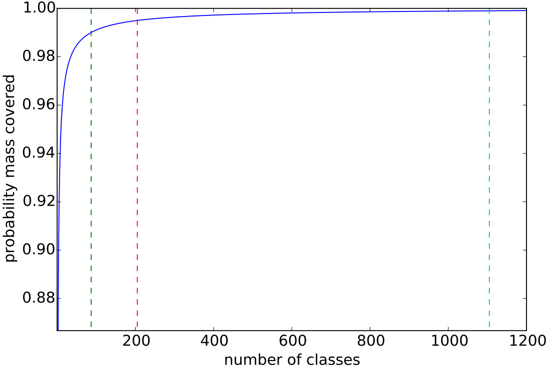
- Very large data set, 309 hours of speech, 18 GB.
- 31×41 inputs, 9000 output classes.

How to do compression with Switchboard

- Very large data set, 309 hours of speech, 18 GB.
- 31×41 inputs, 9000 output classes. $\rightarrow$ Predictions would take 3.6 TB.

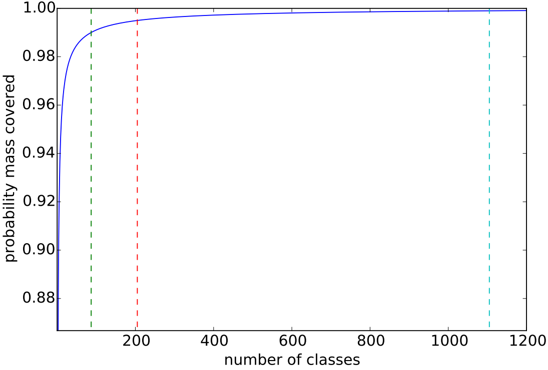
How to do compression with Switchboard

$$\mathrm{M}(C) = \frac{1}{|\{\mathbf{x}_i\}|} \sum_{\mathbf{x}_i} \sum_{y \in \mathrm{TOP}_C(\mathbf{x}_i)} p(y|\mathbf{x}_i).$$



How to do compression with Switchboard

$$\mathrm{M}(C) = \frac{1}{|\{\mathbf{x}_i\}|} \sum_{\mathbf{x}_i} \sum_{y \in \mathrm{TOP}_C(\mathbf{x}_i)} p(y|\mathbf{x}_i).$$



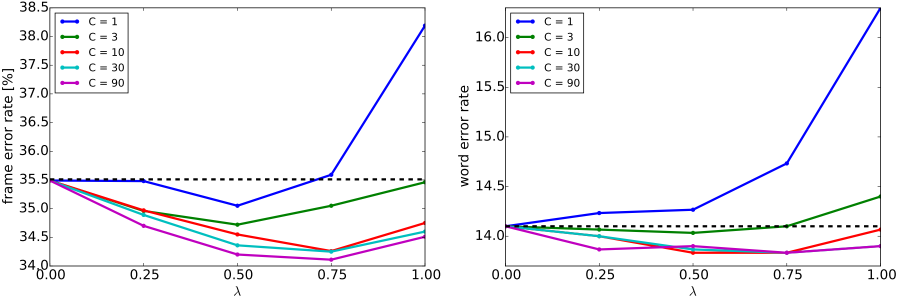
We only keep predictions for classes covering 99% probability mass, we truncate after 90 classes.

Blending LSTMs into CNNs

$$L(\lambda) = \lambda \left[-\sum_j \sum_c p(c|\mathbf{x}_j) \log q(c|\mathbf{x}_j) \right] + (1-\lambda) \left[-\sum_j \log q(y_j|\mathbf{x}_j) \right]$$

BLENDING LSTMS INTO CNNS

$$L(\lambda) = \lambda \left[-\sum_j \sum_c p(c|\mathbf{x}_j) \log q(c|\mathbf{x}_j) \right] + (1-\lambda) \left[-\sum_j \log q(y_j|\mathbf{x}_j) \right]$$



Results

	FER	**WER**	model size	execution time
Sainath et al.-style CNN	37.93%	15.5	$\approx$ 75M	$\times$ 0.75
vision-style CNN	35.51%	14.1	$\approx$ 75M	$\times$ 1.0
LSTM	34.15%	14.4	$\approx$ 65M	$\times$ 5.8
LSTM $\rightarrow$ CNN blending	**34.11%**	**13.83**	$\approx$ 75M	$\times$ 1.0

Summary of model blending

Summary of model blending

- Speech recognition can be improved a lot by vision-style CNNs.

Summary of model blending

- Speech recognition can be improved a lot by vision-style CNNs.
- RNNs and CNNs learn different aspects of the data.

Summary of model blending

- Speech recognition can be improved a lot by vision-style CNNs.
- RNNs and CNNs learn different aspects of the data.
- Recurrent networks for speech recognition may not need to be recurrent.

Summary of model blending

- Speech recognition can be improved a lot by vision-style CNNs.
- RNNs and CNNs learn different aspects of the data.
- Recurrent networks for speech recognition may not need to be recurrent.
- Only "dim knowledge" necessary.

Self-compression

- Train model A.

Self-compression

- Train model A.
- Train an indentical model B, mimicking model A with $\lambda = 0.5$.

Self-compression

- Train model A.
- Train an indentical model B, mimicking model A with $\lambda = 0.5$.
- Model B is more accurate than model A in FER!
  - FER: $35.51 \rightarrow 34.61$.
  - WER: $14.1 \rightarrow 14.1$.

Thank you!

Do Deep Convolutional Nets Really Need to be Deep (Or Even Convolutional)? ICLR 2017

Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana,

Abdel-rahman Mohamed, Matthai Philipose, Matthew Richardson

Blending LSTMs into CNNs. ICLR 2016

Krzysztof J. Geras, Abdel-rahman Mohamed, Rich Caruana, Gregor Urban, Shengjie Wang, Ozlem Aslan,

Matthai Philipose, Matthew Richardson and Charles Sutton