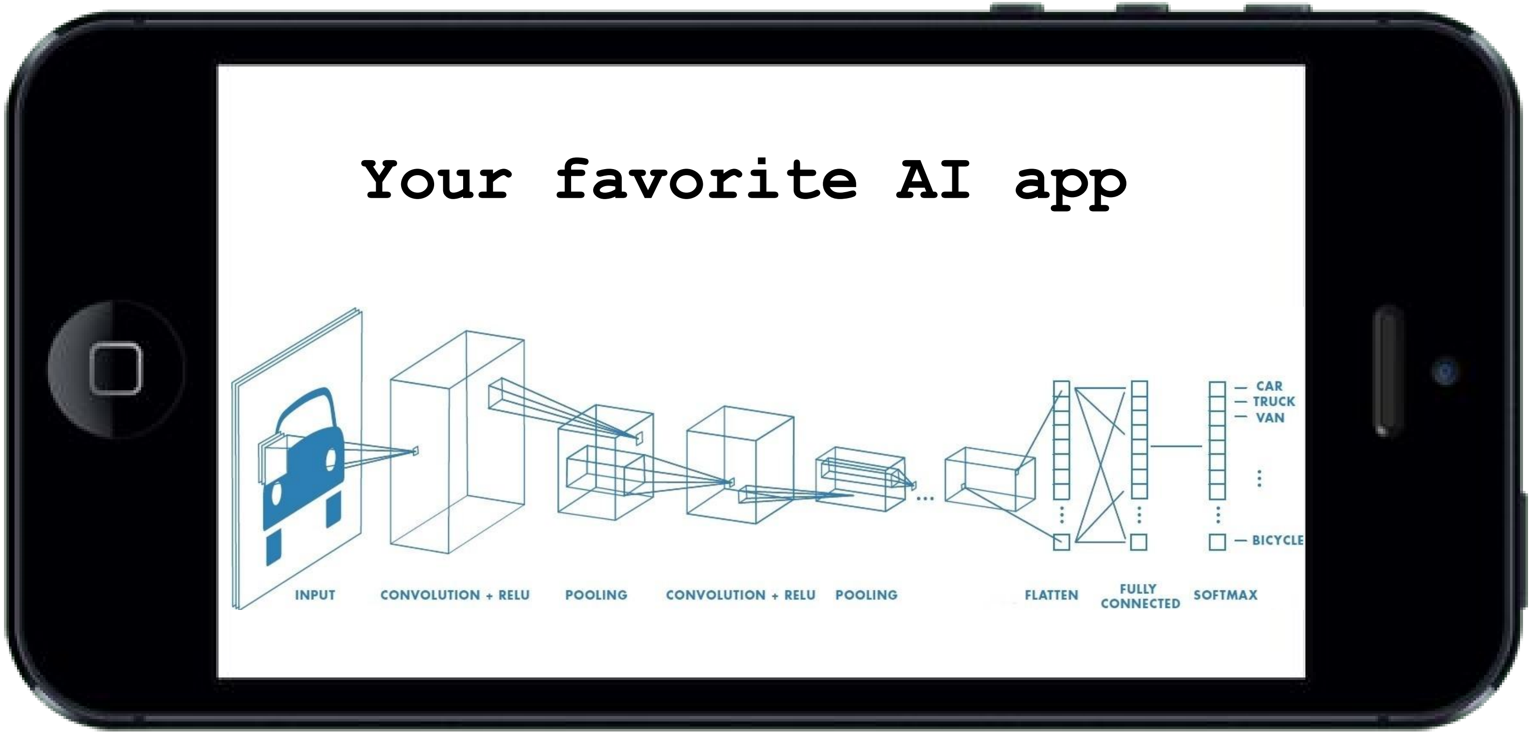
Deep Compression

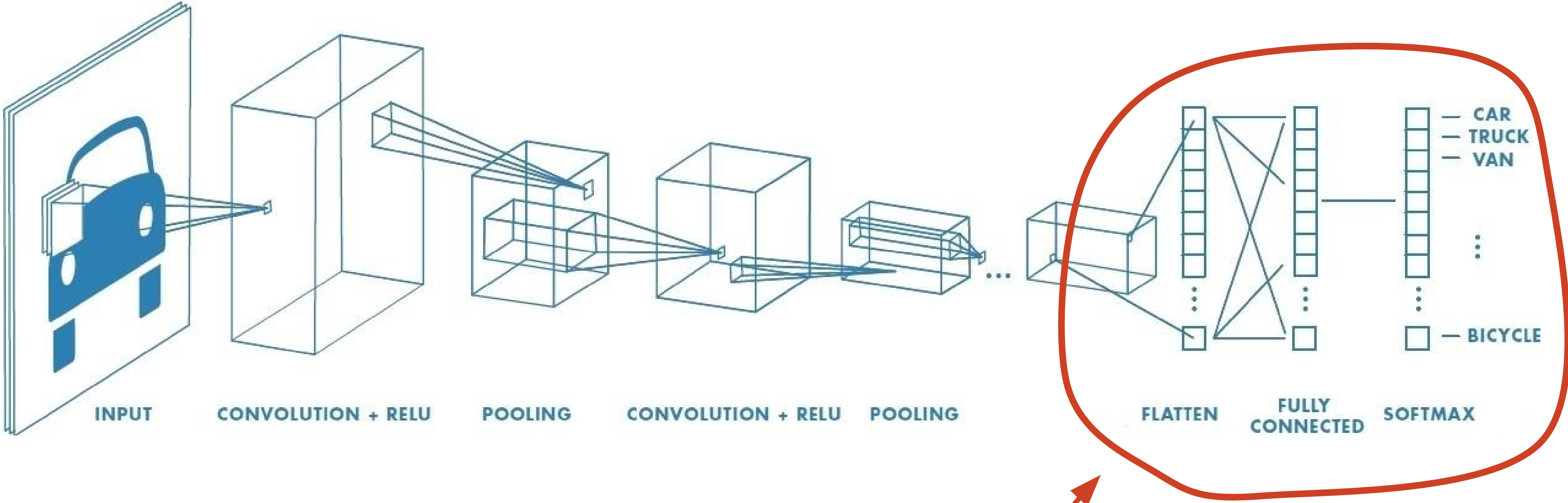
Compressing Deep Neural Networks

Anton Pankratov
Nikita Gryaznov
Yuri Tavirikov

Will it fit?


Your favorite AI app
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
CAR
TRUCK
VAN
BICYCLE

Key Insight



These guys are
90% of memory

How bad is it?

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Operations: $O(n^2)$

Memory: $O(n^2)$

Structures for help

— Maybe we structure our weights?

— Let’s try **circulant matrices**

$$\mathbf{R} = \begin{bmatrix} r_0 & r_{d-1} & \cdots & r_2 & r_1 \\ r_1 & r_0 & \cdots & r_3 & r_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{d-1} & r_{d-1} & \cdots & r_1 & r_0 \end{bmatrix}$$

Does it help?

$$\mathbf{R}\mathbf{x} = \mathbf{r} \circledast \mathbf{x}$$

FFT to the rescue

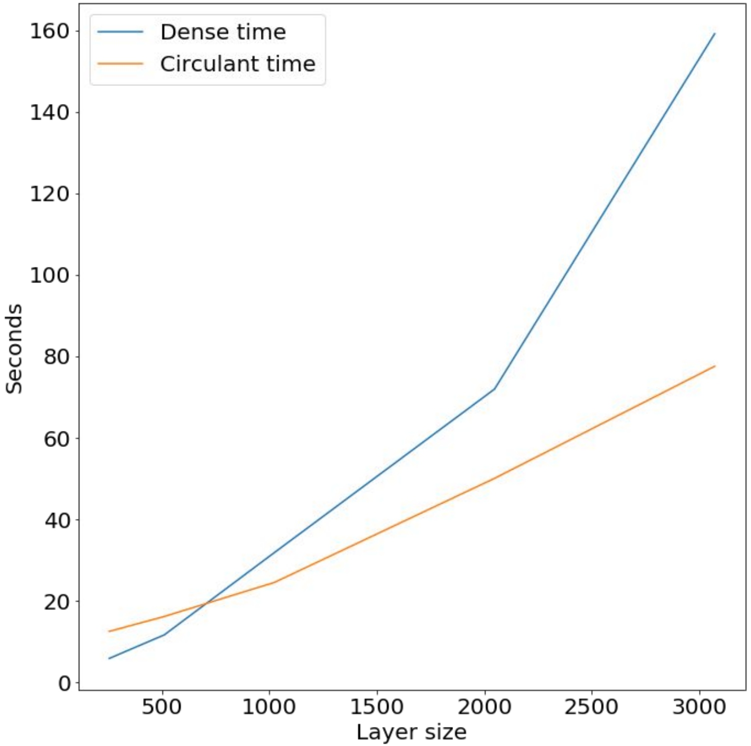
Operations: $O(n \log(n))$

Memory: $O(n)$

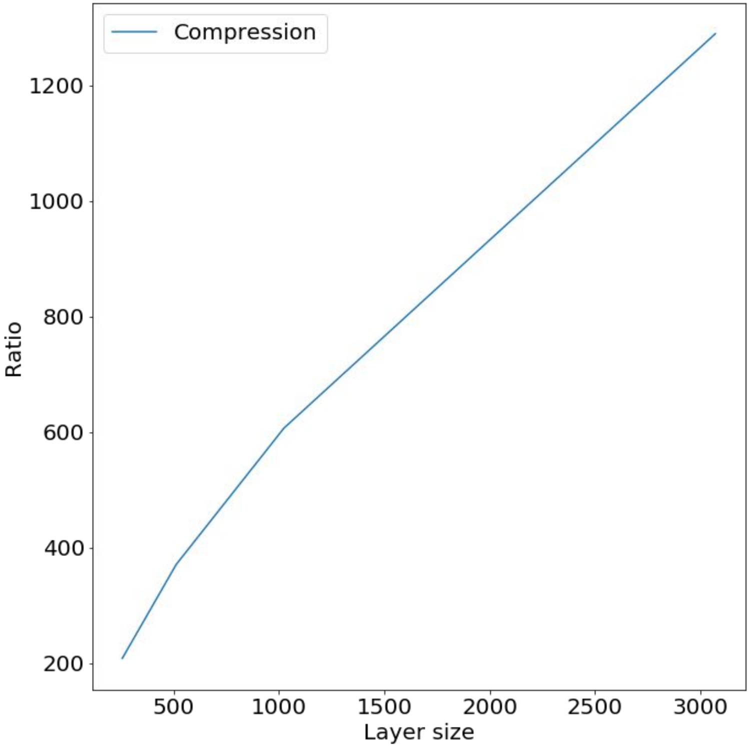
Backward pass is $O(n \log(n))$ too!

Reality

Train time



Compression



Results

MNIST

	Accuracy	Time	Compression
Dense	0.98	211	1
Circulant	0.92	318	370

MNIST Fashion

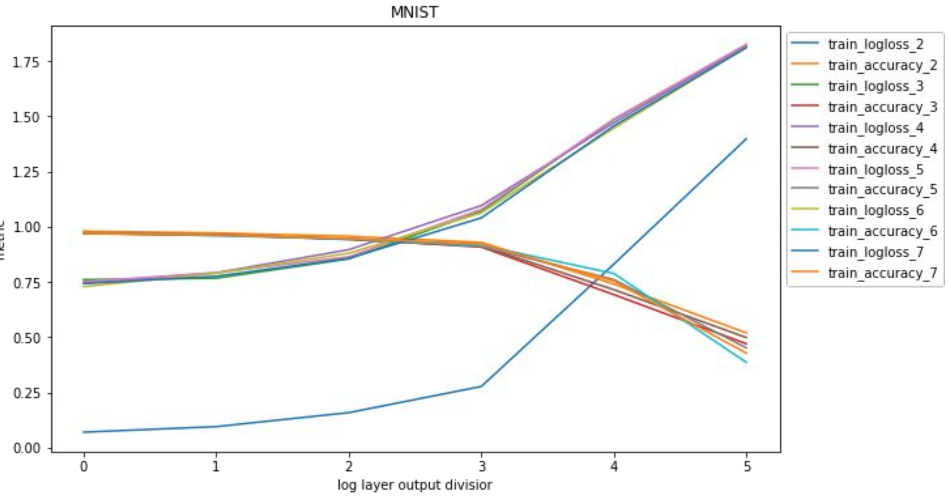
	Accuracy	Time	Compression
Dense	0.89	213	1
Circulant	0.93	352	235

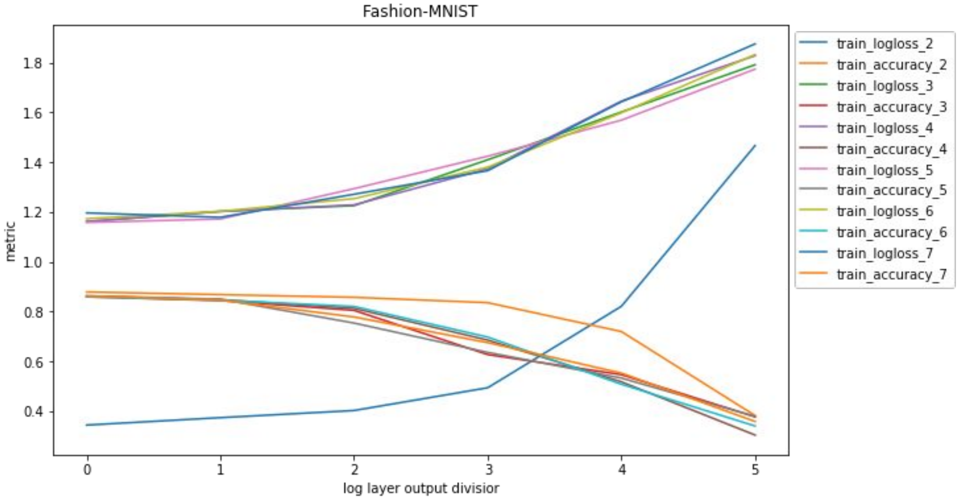
Knowledge distillation

Another idea: train smaller network on outputs of a larger one (not on targets)

Experiment: take an NN and decrease number of output units by 2^k

Knowledge distillation





Results: we can decrease number of weights by a factor of 2-4 without significant loss

SVD approximation

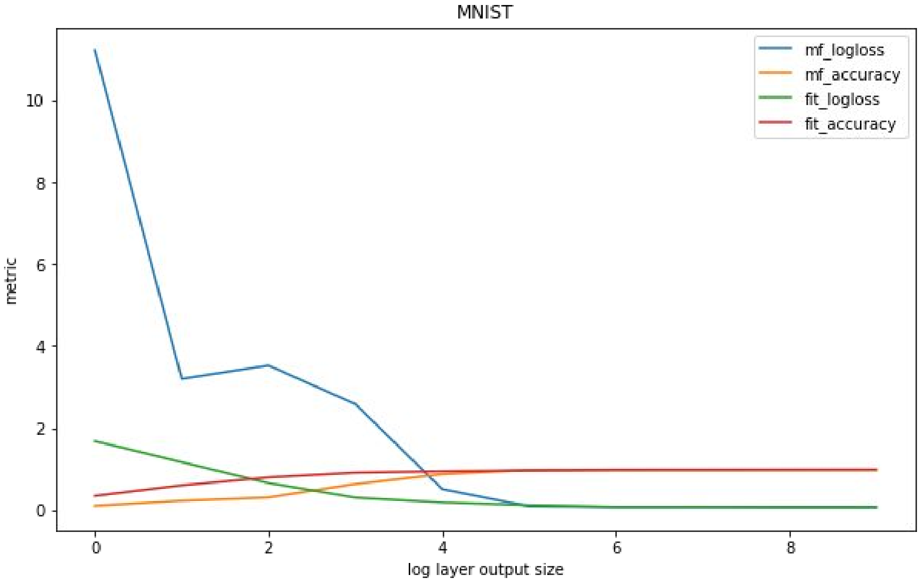
Idea: take dense layer `In x Out`, replace with two dense layers `In x mid + mid x Out` (first has linear activation).

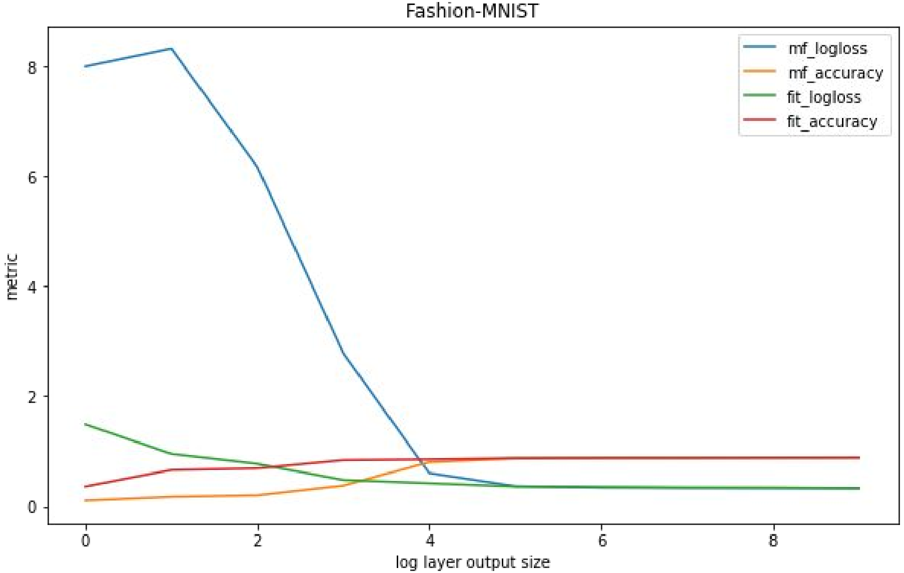
$$W = U\Sigma V^*$$

$$W_1 = \Sigma_r^{\frac{1}{2}} V_r^* \qquad W_2 = U_r \Sigma_r^{\frac{1}{2}}$$

Experiment: take an NN, replace largest dense with two. Compare with same NN tuned from zero.

SVD approximation





Results: about 40x compression can be achieved by losing 2-3%. Plus calculation speedup (from multiplying by `In x Out` now two multiplies `In x mid` and `mid x Out`)

Thank you