# ERL PML
# Deep Learning meeting
# 08/24/2016

Artem Molchanov

# Overview

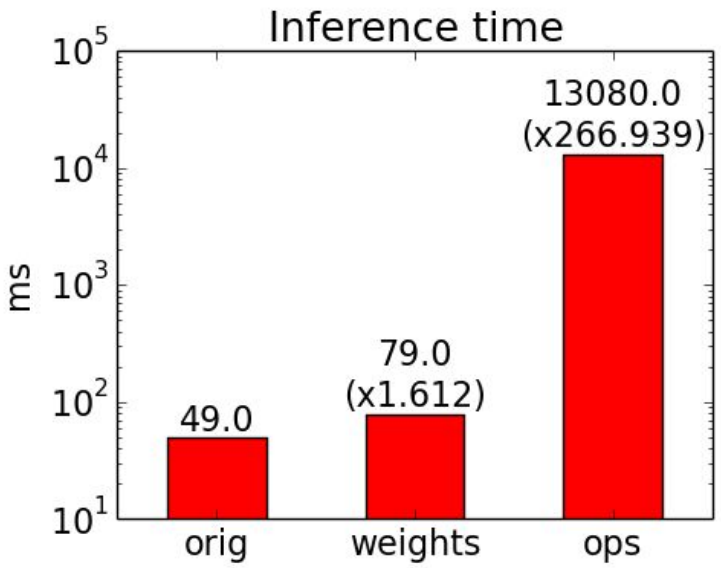
- **Compression**
  - *Quantization*
    - 8-bit with ops results
  - *Pruning*
    - Fully connected layers: Model
    - Fully connected layers: Results
    - Sparse CNN re-running
- **Intermediate feature learning**
  - *Max/Avg pooling features*
- **Discussion**

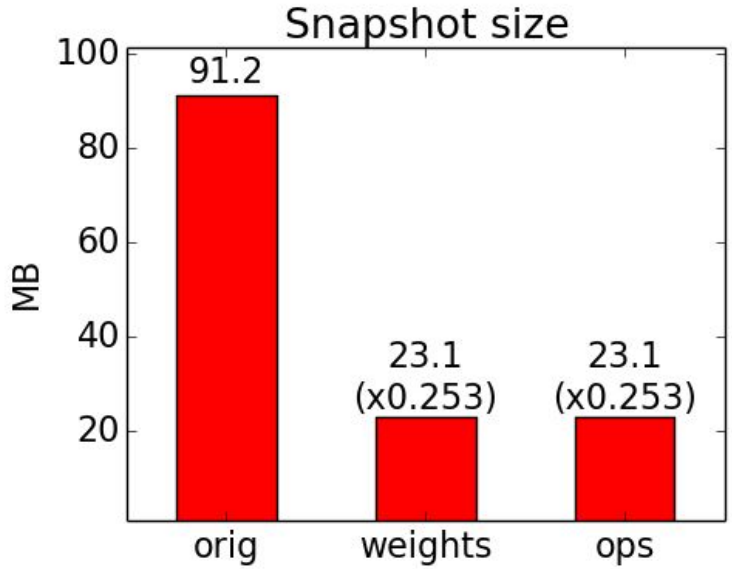
# Quantization: Results for Inception V3

- **Inception V3 net**

- **Legend:**
  - **Orig -** no quantization
  - **Weights** - weight quantization (dequantize at runtime)
  - **Ops** - quantized ops and weights

## 8bit quantization results

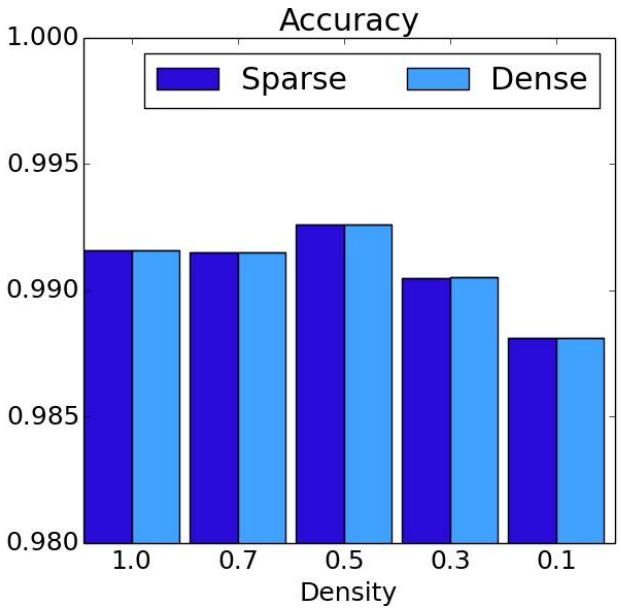


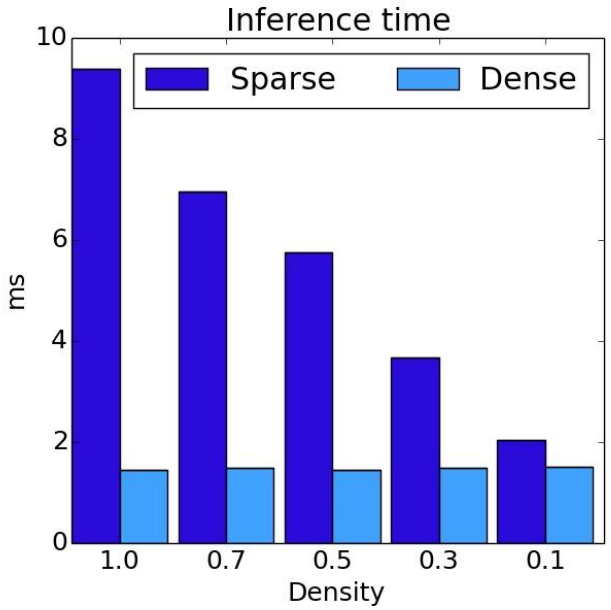



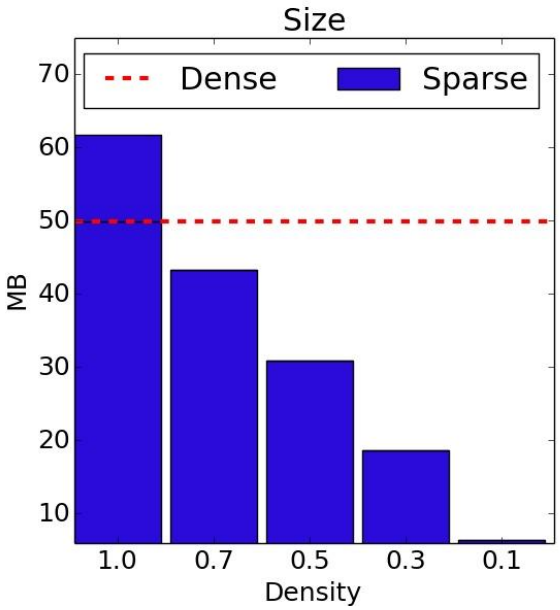
# Pruning: FCN. MNIST model

- **Simple MNIST model:**
  - 2 convolutional layers
  - 2 fully connected layers
  - Softmax
- **FC implementation**:
  - Dense:
    - *tf.**matmul**(activations, w_dense)*
  - Sparse:
    - *h_mult = tf.**sparse_tensor_dense_matmul**(w_sparse, activations, adjoint_a=True, adjoint_b=True)*
      *h_mult_tr = tf.transpose(h_mult)*
- **Re-Training implementation (to keep weights == 0):**
  - Gradient masking (see example)
- **Tool:**
  - With no effort can prune *.ckpt (*.pb) model w/o retraining.
  - Retraining is task specific: add gradient masking to your implementation
  - Independent sparse model is required (see example)

# Pruning: FCN results



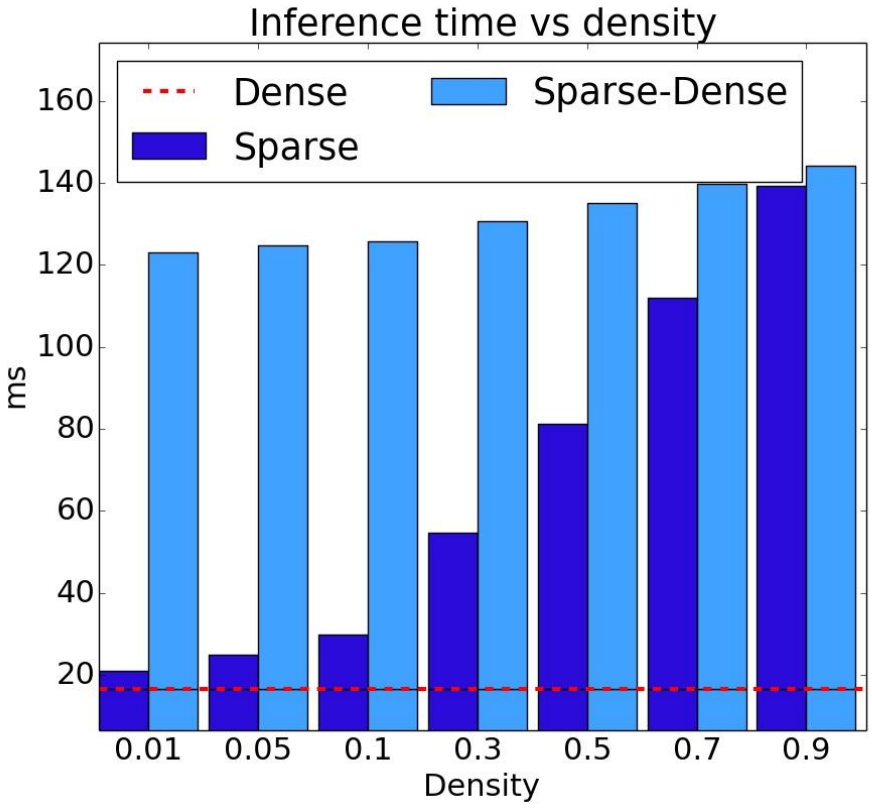

- **Results:**
  - Minor loss after pruning and retraining
  - Accuracies are equivalent for sparse/dense
  - Linear improvement in performance
  - Linear decrease in size
  - **Dense model outperforms the sparse one !**

# Pruning: sparse CNN re-run

**Sparse-Dense** == convolution re-implemented using:

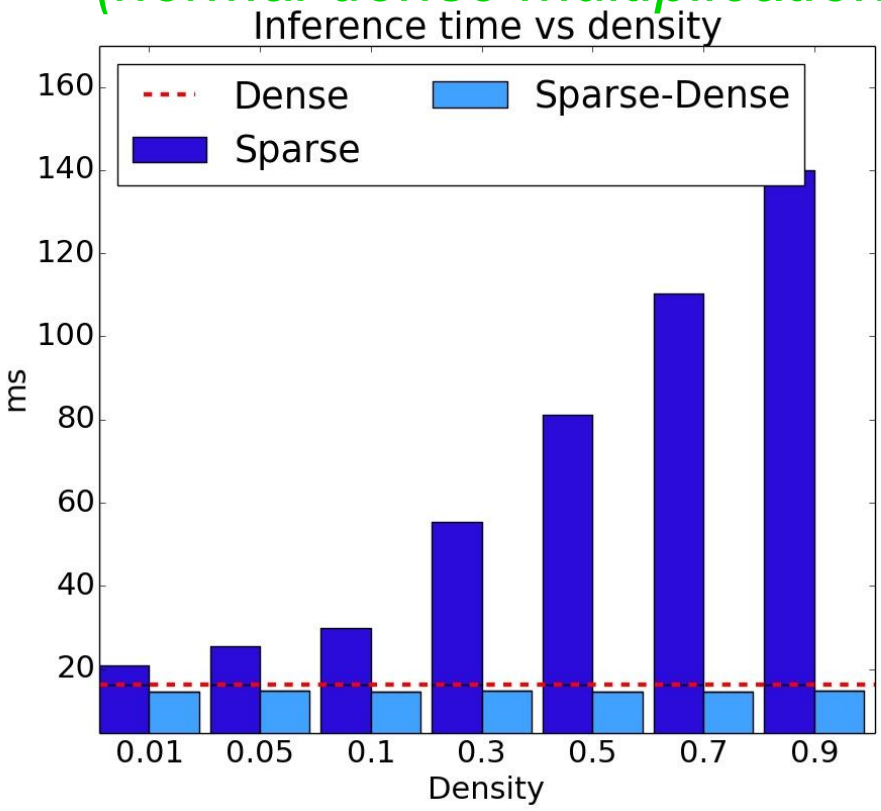
- ***tf.matmul****(activations, W,* ***b_is_sparse****)*

***b_is_sparse = True***




***b_is_sparse = False***

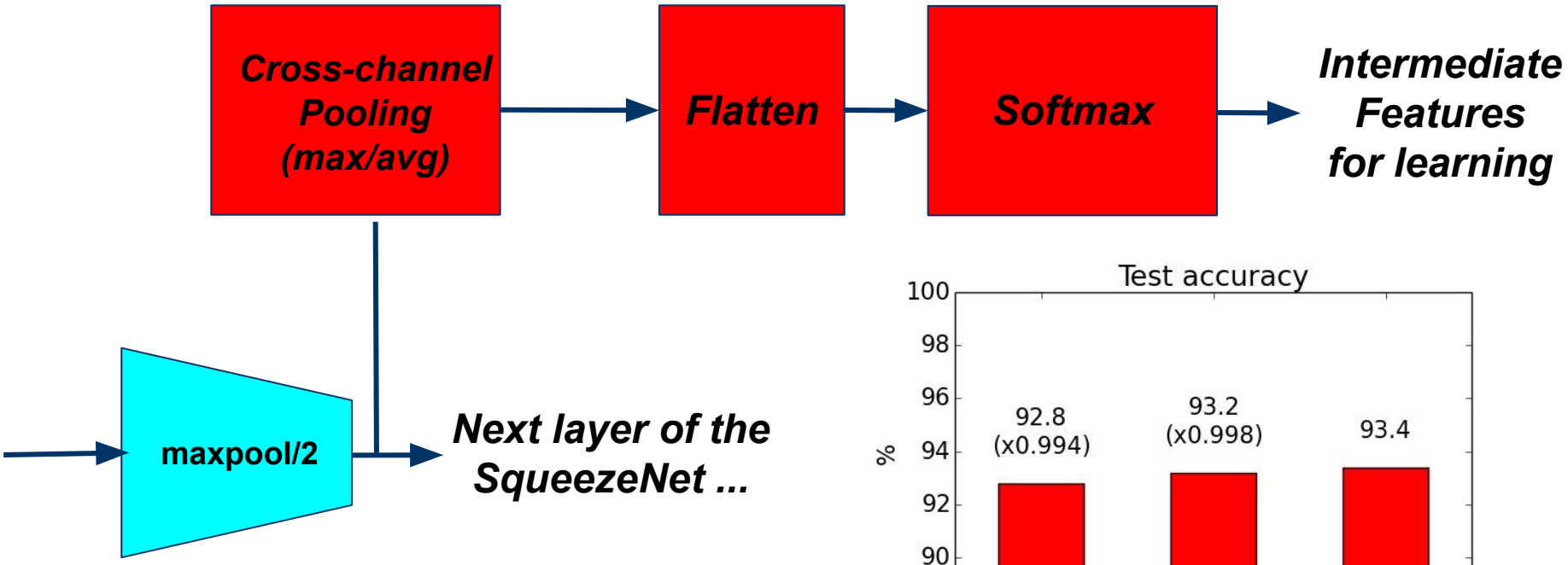
*(normal dense multiplication)*




ERL

# Intermediate Features: Max/Avg pooling

- *Intermediate features extracted from **SqueezeNet** and learned by **SqueezeNetX8** (8-times more narrow)*
- *Extraction is done for every maxpooling layer (i.e. 3 sets of features)*



# Compression: Lessons learned

- **Quantization:**
  - Linear decrease in size (x4 for 8 bit)
  - Inference time increases 20-60 %
  - No loss in accuracy
- **Pruning:**
  - Almost linear decrease in size (up to x10 for FC)
  - Increase in inference time due to inefficient implementation of sparse operations in TF
  - Minor loss in accuracy
- **Model reduction with Distillation**:
  - Better than linear decrease in size for convolutions
  - Decrease in inference time (30-40% for segmentation)
  - Minor loss in accuracy

# THANKS !

## THANKS TO
## THE WHOLE ERL TEAM !