Motivation

- Computational Efficiency (inkl. Horowitz) !! Why do we do this?
- Improved Generalization (train with low precision (lp) but keep hp weights and use them during test -¿ better results) Some power gains during training, no power gains during test.
- Energy efficiency during test time if you use lp weights, sacrifice accuracy for power saving
- Examples and related concepts: dropout, ReLu (sparse gradient), influence of feature discretization,

Binarization Schemes

- Deterministic vs. Stochastic

Gradient Estimation

- Path Derivative Gradient Estimators (reparametrization, ST, SlopeAnnealing)
- Score Function based Gradient Estimators (reinforce+many others)
- Expectation Backpropagation

See Gumble Softmax Paper for overview

Algorithms Adam + BN

- General Review
- Illustrate their importance for QNNs
- Implement algorithm to see how it works without Adam and BN?

Full Algorithm

- Try to visualize, illustrate procedure (highlight similarities and differences to conventional NN training and inference)

Experiments

- Describe different NN architectures (DNN, CNN, RNN)
- Describe data sets (MNIST, CIFAR, ImageNet, PennTree)
- binary vs low precision

Results

- Training results
- Performance results (look up older benchmarks to compare loss in in accuracy to)
- find comparisons from other references

Evolution of the QNN paper

Training NN with lp multiplications

- The inital motivation was to reduce the power consumption of multipliers, which are claimed to be the most power hungry.
- A common pattern already shows: lp multipliers + hp accumulators. Parameter updates are also hp.
- Cost of multiplier $O(precision^2)$ while cost of accumulator $O(precision)$
- Problem with fixed point: low dynamic range $\rightarrow$ dynamic fixed point.
- Evaluation of maxout on MNIST, CIFAR, SVHN with three formats (floating point, fixed point, dynamic fixed point)
- Techniques: Maxout, dropout, momentum, weight decay, dyn. fixed point, update prc vs prop prec.
- Comparison floating point vs. fixed point ?
- Explain demand for sufficient precison of update (compared to Propagations) due to SGD updates
- Half-precision has little to no impact (hp fine-tuning?)
- Couple of plots showing final test error as function of everything
- References to lp network training in the 90ies!
- No evaluation of computational gains, only show how robust NNs are to lp muls.

Binary Connect: Training DNNs with binary weights during propagations

- Emphasis shifts from