Deep Learning for Speech Recognition

Piotr Ambroszczyk   Łukasz Kondraciuk

Wojciech Przybyszewski   Jan Tabaszewski

Advisor: Janina Mincer-Daszkiewicz PhD

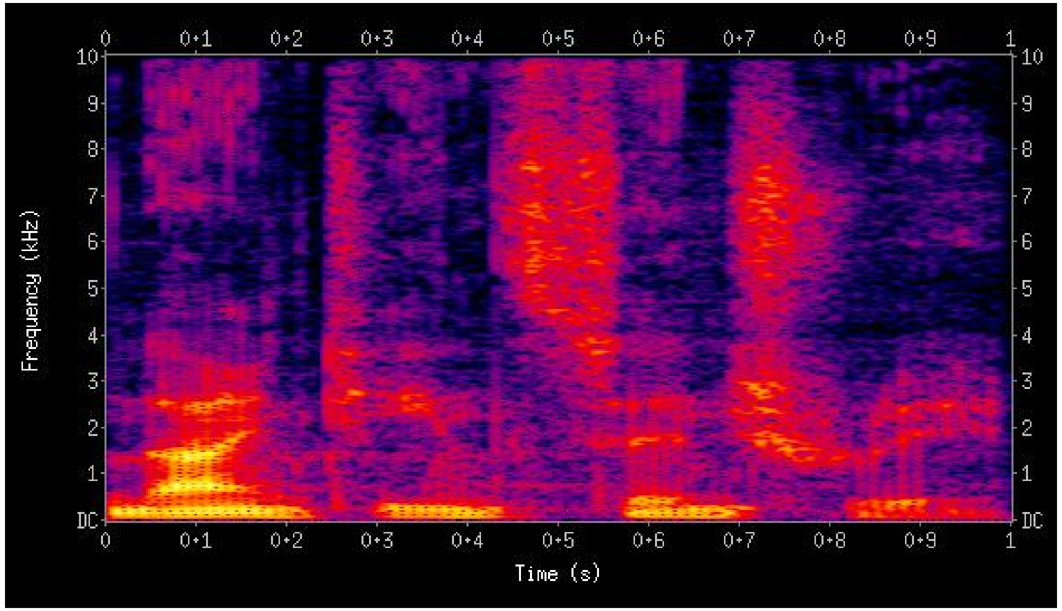
with NVIDIA Corporation

What is speech recognition?

- Recognition and translation of spoken language into text by computers
- Crucial problem for many areas of modern technology industry
- Communicating with electronic devices by talking to them
- Before - solutions which use complex algorithms and fine-tuned parameters
- Now - fully deep learning approach
- In our thesis we concentrate on Deep Speech 2 neural net model (published by Baidu) and experiment with it

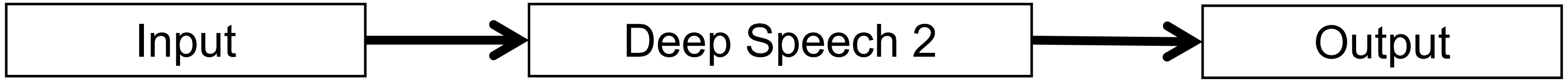
Formal specification of the problem

Dataset - set of pairs (input, output)



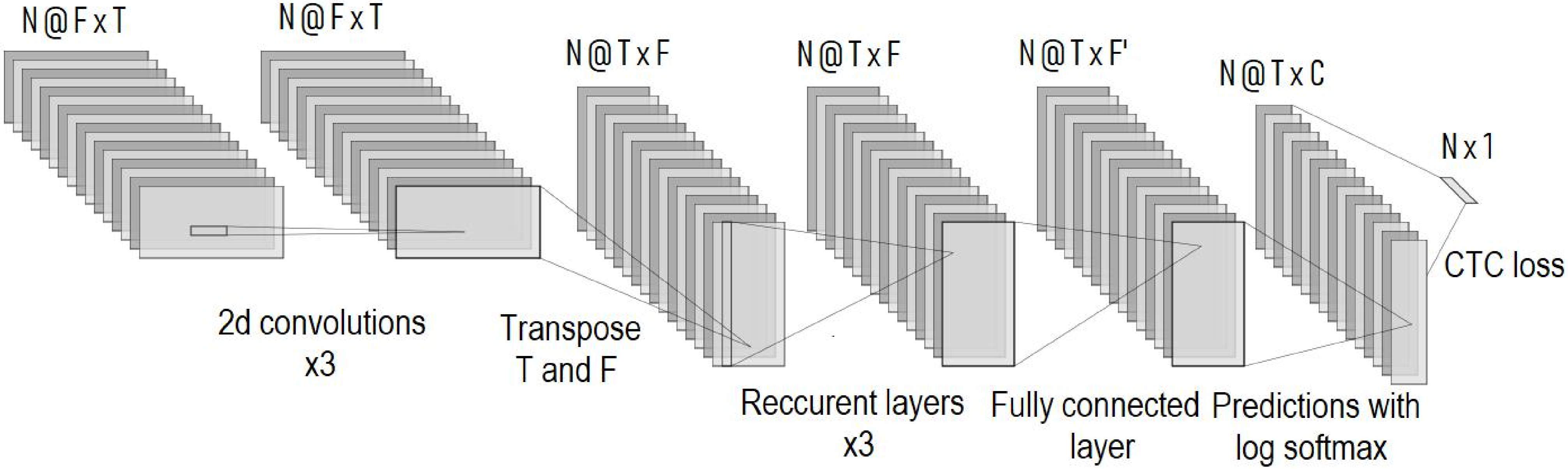
Picture from Wikipedia

to be or not to be
that is the question ...


Input → Deep Speech 2 → Output

Model architecture

- Combination of convolutional, recurrent and fully connected layers



Loss function - Connectionist Temporal Classification

- Loss function designed for tasks where the timing is variable (and length of the output is not a function of length of the input)
- Special symbol blank added to our alphabet

mary -> mary, marry, mmmmaaarry, m__marr_r_r_y_y_y

marry -> mar_ry, mmmaa_arrr_r_rr_ryy, marrrrr_ryy_y

Language model

- We decided to use 4grams

4-gram	Probability
to be or not	0.00001%
milk computer dance me	0%
lambda calculus is the	0.00000003%
to be or nein	0%

Generating transcriptions

- Neural net returns matrix TxC where C is number of symbols in our alphabet and T is time series length. Each cell of this matrix contains probability of a given symbol in a given timestamp
- In the formula below y is a transcription and x is an output of our model

$$Q(y) = \log\left(\mathbb{P}_{ctc}(y|x)\right) + \alpha \cdot \log\left(\mathbb{P}_{lm}(y)\right) + \beta \cdot word\_count(y)$$

- We use beam search algorithm to generate final transcriptions

Measuring results - Word Error Rate

- For measuring quality of our results we use Word Error Rate (WER) metric

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

- S – the number of substitutions,
- D – the number of deletions,
- I – the number of insertions,
- C – the number of correct words,
- N – the number of words in the reference ($N = S + D + C$).

Model output	Real transcription	WER
twil you forgiti me now	will you forgive me now	40%
tfer welm ma am	farewell madam	200%

Results

Model	Dataset size	WER
Baidu	12000h	8.5%
Baidu	1200h	13.8%
Mozilla	~3000h	6.5%
Our model	960h	10.4%
Native speaker	whole life	~5.8%

Audio examples

Our model	Transcription	WER

Audio examples

Our model	Transcription	WER
AS A MATTER OF FACT HE COULD NOT SAID SOLMS FOR I ENTERED BY THE SIDE DOOR		

Audio examples

Our model	Transcription	WER
AS A MATTER OF FACT HE COULD NOT SAID SOLMS FOR I ENTERED BY THE SIDE DOOR		
YOU HEAR WHAT SIR FERDINAND A BROWN HA SAT REPLIED CAPTAIN BATLEAX		

Audio examples

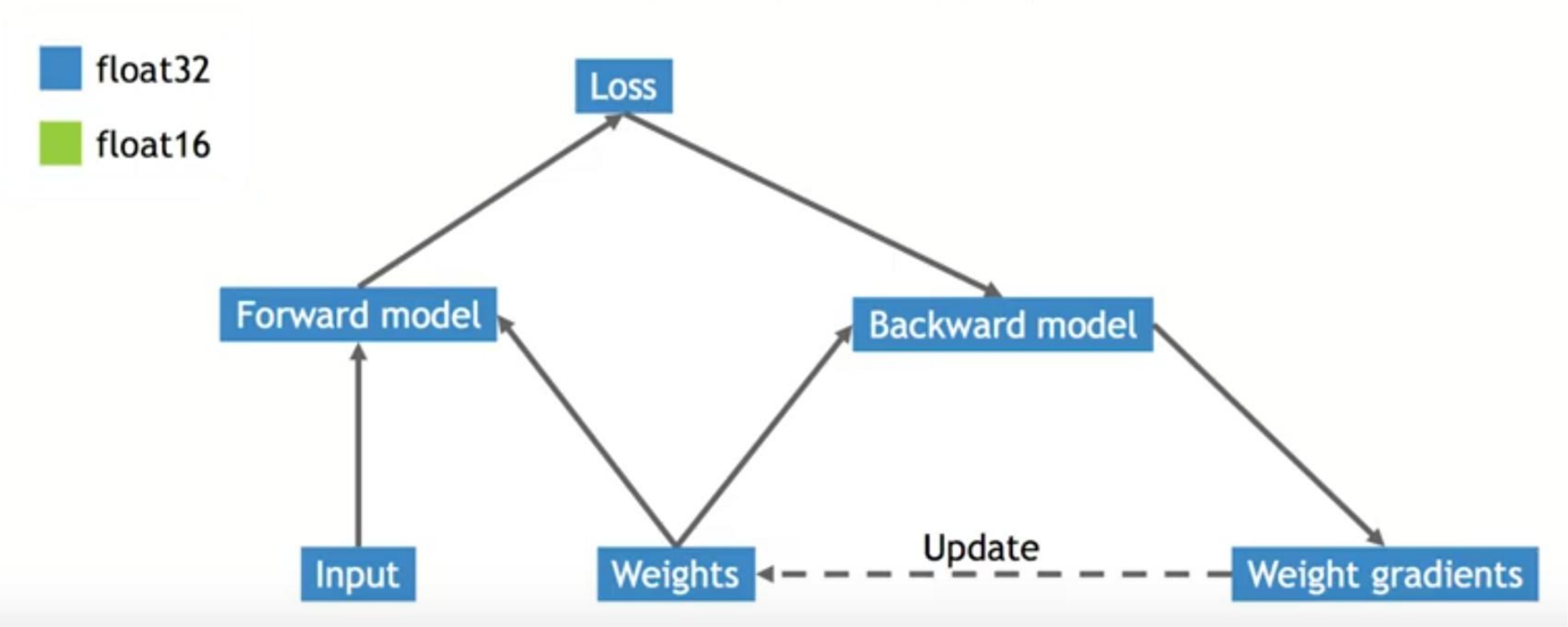
Our model	Transcription	WER
AS A MATTER OF FACT HE COULD NOT SAID SOLMS FOR I ENTERED BY THE SIDE DOOR		
YOU HEAR WHAT SIR FERDINAND A BROWN HA SAT REPLIED CAPTAIN BATLEAX		
THE ARMY SOUND THE PEOPLE IN POVERTY AND LEFT THEM IN COMPARATIVE WEALTH		

Audio examples

Our model	Transcription	WER
AS A MATTER OF FACT HE COULD NOT SAID SOLMS FOR I ENTERED BY THE SIDE DOOR	AS A MATTER OF FACT HE COULD NOT SAID SOAMES FOR I ENTERED BY THE SIDE DOOR	6%
YOU HEAR WHAT SIR FERDINAND A BROWN HA SAT REPLIED CAPTAIN BATLEAX	YOU HEAR WHAT SIR FERDINANDO BROWN HAS SAID REPLIED CAPTAIN BATTLEAX	45%
THE ARMY SOUND THE PEOPLE IN POVERTY AND LEFT THEM IN COMPARATIVE WEALTH	THE ARMY FOUND THE PEOPLE IN POVERTY AND LEFT THEM IN COMPARATIVE WEALTH	8%

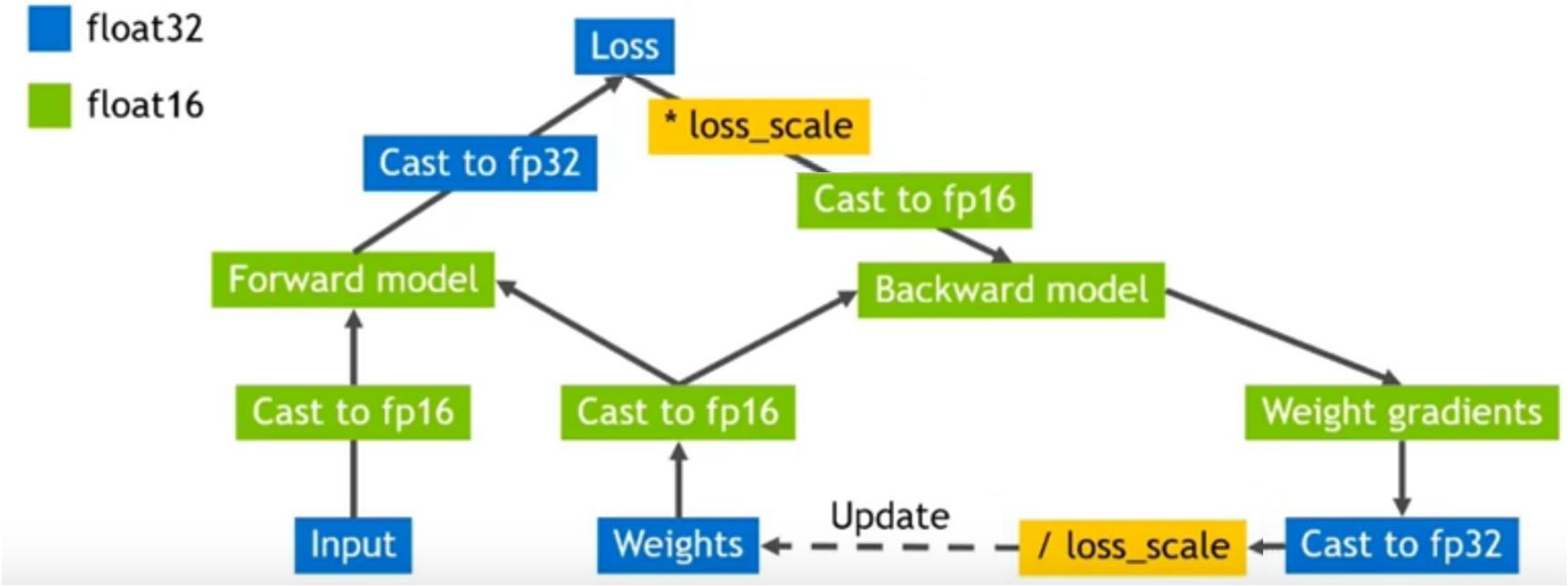
Experiments

Usual training process

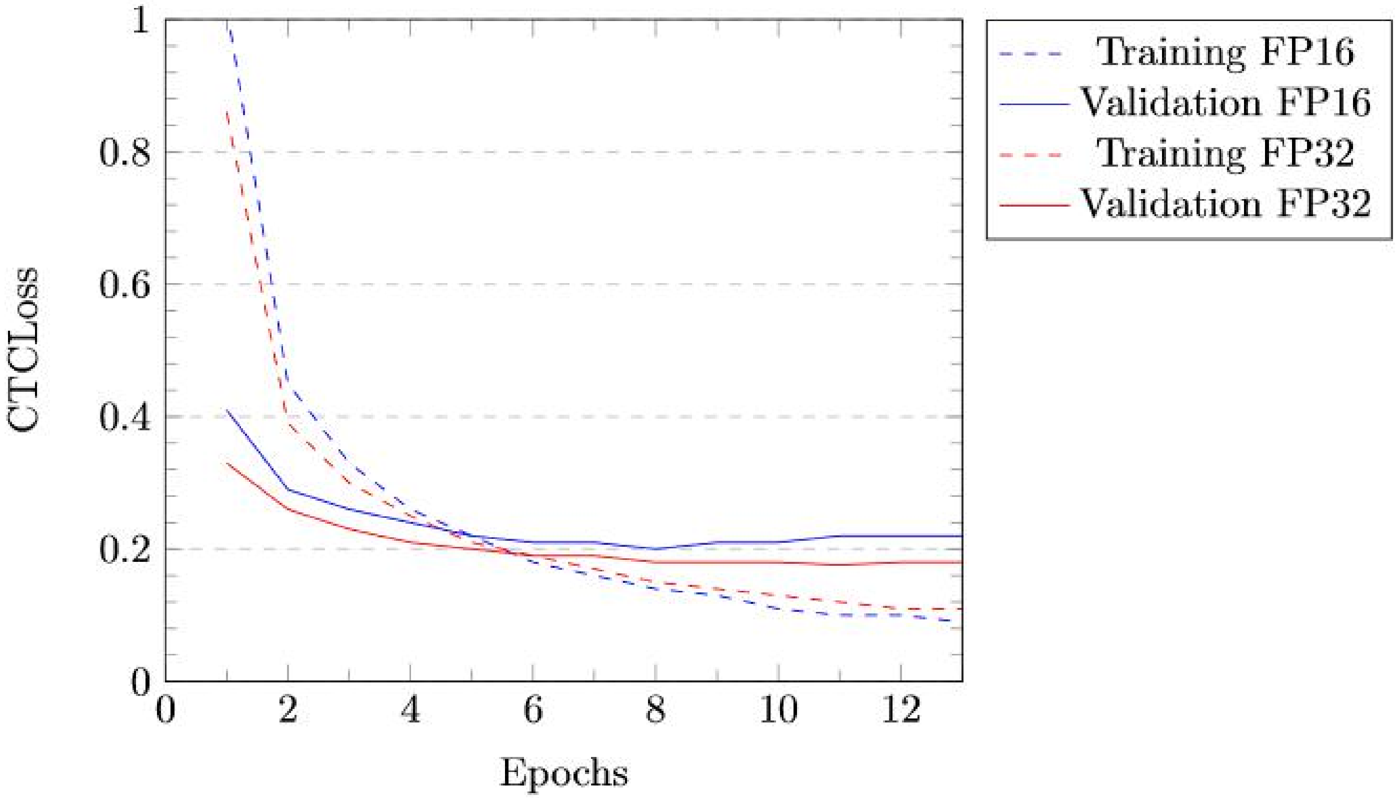


Picture from NVIDIA developer YouTube channel

Mixed precision training



Picture from NVIDIA developer YouTube channel



Epoch on fp32 takes around 135 minutes

Epoch on fp16 takes around 75 minutes

Multi-GPU scaling

Number of GPUs	Time per one epoch	Speedup
1	118 minutes	1
2	75 minutes	1.57
4	35 minutes	3.37

Ablation study

- Batch normalization
- Dropout
- L2 regularization
- Reccurent unit type
- Sortagrad
- Xavier initialization

Thank you for your attention!

Special thanks to our supervisor
Janina Mincer-Daszkiewicz PhD
and NVIDIA Corporation with our mentor
Grzegorz Karch PhD