Бинаризована неуронска мрежа

Шука Александар

RT-RK
COMPUTER BASED SYSTEMS

MACHINE LEARNING AND
APPLICATIONS GROUP

Преглед

- Уопштено о проблему
- Квантизована неуронска мрежа
- Доприноси рада
- Функција бинаризације
- Пропагација градијената кроз дискретизацију
- Израчунавање и акумулација градијената
- Shift-based Batch Normalization
- Shift-based AdaMax
- Скаларни производ помоћу XNOR i popcount
- Резултати

Уопштено о проблему

- Традиционално, дубоке неуронске мреже (ДНН) користе вредности у покретном зарезу (32 бита).

Уопштено о проблему

- Традиционално, дубоке неуронске мреже (ДНН) користе вредности у покретном зарезу (32 бита).
- Веровало се да су вредности у покретном зарезу потребне због тачности.

Уопштено о проблему

- Традиционално, дубоке неуронске мреже (ДНН) користе вредности у покретном зарезу (32 бита).
- Веровало се да су вредности у покретном зарезу потребне због тачности.
- Највећи број операција током рада са неуронским мрежама (рачунање тежинских сума) своди се на рачунање скаларних производа.

Уопштено о проблему

- Рачунање скаларног производа састоји се од низа Multiply-Accumulate (MAC) операција.

Уопштено о проблему

- Рачунање скаларног производа састоји се од низа Multiply-Accumulate (MAC) операција.
- Хардвер за општу намену је неефикасан када је у питању масиван број MAC операција, тако да се ДНН искључиво тренирају на брзим и захтевним GPU

Уопштено о проблему

- Шта се дешава данас, када имамо велики број наменских уређаја мале снаге (телефони, сатови, дронови, наочаре...).

Уопштено о проблему

- Шта се дешава данас, када имамо велики број наменских уређаја мале снаге (телефони, сатови, дронови, наочаре...).
- Већина њих су ограничени у погледу величине, снаге и меморијског капацитета.

Уопштено о проблему

- Шта се дешава данас, када имамо велики број наменских уређаја мале снаге (телефони, сатови, дронови, наочаре...).
- Већина њих су ограничени у погледу величине, снаге и меморијског капацитета.
- Да би ублажили проблем, људи почињу са истраживањем ДНН које користе активације и тежине са малом прецизношћу.

Квантизована неуронска мрежа

- Основна идеја је та да је опште познато да неуронске мреже добро подносе шум, па смањење прецизности може да се гледа као увођење додатног шума.

Квантизована неуронска мрежа

- Основна идеја је та да је опште познато да неуронске мреже добро подносе шум, па смањење прецизности може да се гледа као увођење додатног шума.
- Истраживања предлажу технику која се назива Квантизована неуронска мрежа (КНН), која квантизује активације и тежине током тренинга и закључивања.

Квантизована неуронска мрежа

- Све MAC операције се могу заменити са XNOR и population count операцијама.

Квантизована неуронска мрежа

- Све MAC операције се могу заменити са XNOR и population count операцијама.
- Посебно корисно код КНН са екстремно ниском прецизношћу, на пример 1 бит по тежини и активацији што нас доводи до Бинарних неуронских мрежа (БНН).

Квантизована неуронска мрежа

- Све MAC операције се могу заменити са XNOR и population count операцијама.
- Посебно корисно код КНН са екстремно ниском прецизношћу, на пример 1 бит по тежини и активацији што нас доводи до Бинарних неуронских мрежа (БНН).
- Овај метод је најефектнији у случају конволутивних неуронских мрежа код којих је број параметара јако велики.

Аутори

- Itay Hubara*
- Matthieu Courbariaux*
- Daniel Soudry
- Ran El-Yaniv
- Yoshua Bengio

Доприноси рада

- Уводи се метода за тренирање КНН, активације и тежине са ниским прецизностима. У екстремним случајевима користи 1 бит по тежини и активацији (БНН).

Доприноси рада

- Уводи се метода за тренирање КНН, активације и тежине са ниским прецизностима. У екстремним случајевима користи 1 бит по тежини и активацији (БНН).
- Експерименти имплементирани у Thorch7 и Theano, који показују да је могуће истренирати БНН мрежу на MNIST, CIFAR-10 и SVNH скуповима података и постићи скоро state-of-the-art резултате.

Доприноси рада

- Представљени су прелиминарни резултати на квантизираним градијентима и показује се да је могуће користите само 6 бита са само малом деградацијом тачности.

Доприноси рада

- Представљени су прелиминарни резултати на квантизираним градијентима и показује се да је могуће користите само 6 бита са само малом деградацијом тачности.
- Приказују се резултати са Penn Treebank скупом података користећи језичке моделе (vanilla RNN, LSTM), и показује се да са 4-битним тежинама и активацијама рекурентна КНН постиже сличну тачност као 32-битна у покретном зарезу.

Доприноси рада

- Показује се да КНН драстично смањује потрошњу меморије и замењује аритметчке оперције битовским. Као резултат тога очекује се значајно повећање енергетске ефикасности. Такође бинаризована конволутивна НН може довести да понављања бинарног конволутивног кернела, тврди се да би наменски хардвер могао да смањи временску сложеност за 60%.

Доприноси рада

- Показује се да KHH драстично смањује потрошњу меморије и замењује аритметчке оперције битовским. Као резултат тога очекује се значајно повећање енергетске ефикасности. Такође бинаризована конволутивна НН може довести да понављања бинарног конволутивног кернела, тврди се да би наменски хардвер могао да смањи временску сложеност за 60%.
- Такође имплементиран је GPU кернел за множење бинаризованих матрица који омогућава да се MNIST скуп података савлада 7 пута брже него са неоптимизованим GPU кернелом.

Доприноси рада

- Показује се да KHH драстично смањује потрошњу меморије и замењује аритметчке оперције битовским. Као резултат тога очекује се значајно повећање енергетске ефикасности. Такође бинаризована конволутивна HH може довести да понављања бинарног конволутивног кернела, тврди се да би наменски хардвер могао да смањи временску сложеност за 60%.
- Такође имплементиран је GPU кернел за множење бинаризованих матрица који омогућава да се MNIST скуп података савлада 7 пута брже него са неоптимизованим GPU кернелом.
- Код је доступан на интернету.

Алгоритам

{1. Computing the parameter gradients:}
{1.1. Forward propagation:}
for $k = 1$ to L **do**
  $W_k^b \leftarrow \text{Binarize}(W_k)$
  $s_k \leftarrow a_{k-1}^b W_k^b$
  $a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$
  if $k < L$ **then**
    $a_k^b \leftarrow \text{Binarize}(a_k)$
  end if
end for
{1.2. Backward propagation:}
{Note that the gradients are not binary.}
Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L and a^*
for $k = L$ to 1 **do**
  if $k < L$ **then**
    $g_{a_k} \leftarrow g_{a_k^b} \circ 1_{|a_k| \leq 1}$
  end if
  $(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$
  $g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$
  $g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$
end for
{2. Accumulating the parameter gradients:}
for $k = 1$ to L **do**
  $\theta_k^{t+1} \leftarrow \text{Update}(\theta_k, \eta, g_{\theta_k})$
  $W_k^{t+1} \leftarrow \text{Clip}(\text{Update}(W_k, \gamma_k \eta, g_{W_k^b}), -1, 1)$
  $\eta^{t+1} \leftarrow \lambda \eta$
end for

Функција бинаризације

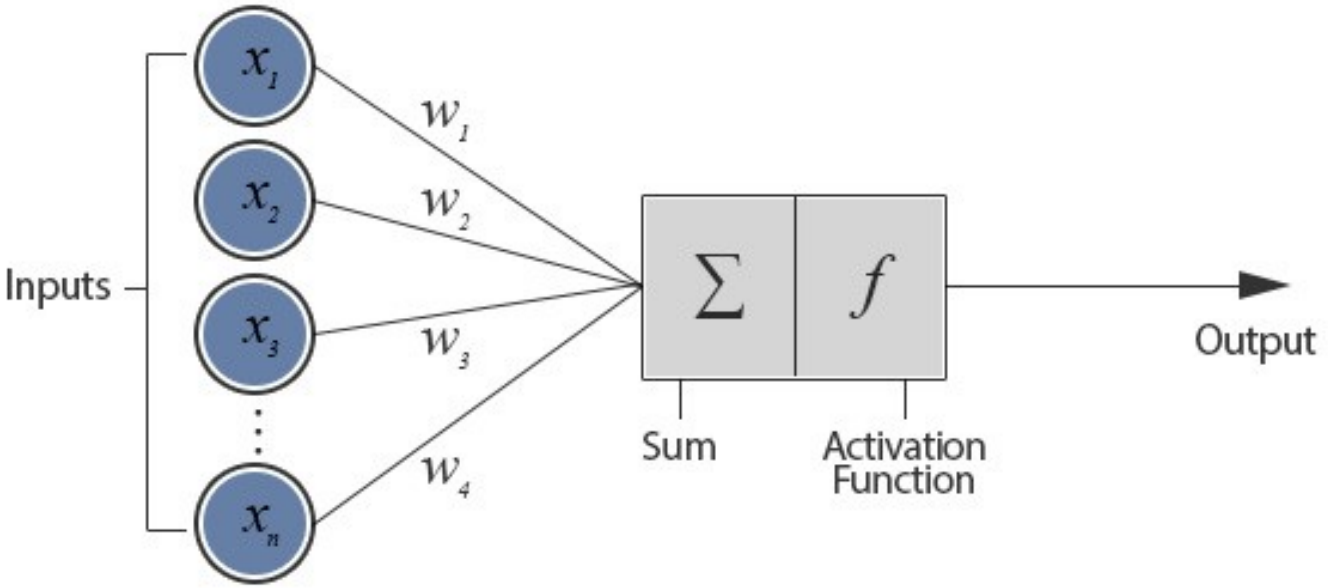
- Трансформисање реалних вредности променљивих у бинарне (+1 и -1).

Функција бинаризације

- Трансформисање реалних вредности променљивих у бинарне (+1 и -1).
- Две функције, детерминистичка и стохастичка.

$$x^b = \text{sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{otherwise,} \end{cases}$$

$$x^b = \begin{cases} +1 & \text{with probability } p = \sigma(x), \\ -1 & \text{with probability } 1 - p, \end{cases}$$



{1.1. Forward propagation:}
for $k = 1$ to L **do**
  $W_k^b \leftarrow \text{Binarize}(W_k)$
  $s_k \leftarrow a_{k-1}^b W_k^b$
  $a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$
  if $k < L$ **then**
    $a_k^b \leftarrow \text{Binarize}(a_k)$
  end if
end for

Функција бинаризације

Пропагација градијената кроз дискретизацију

- Извод sign функције је нула скоро свуда, што је чини неупотребљивом у случају backpropagation алгоритма.

Пропагација градијената кроз дискретизацију

- Извод sign функције је нула скоро свуда, што је чини неупотребљивом у случају backpropagation алгоритма.
- Предлаже се коришћење straight-through естиматора.
- Приликом пропагације уназад на активациону функцију гледа се као на идентитет, чији извод је 1.

$$q = \mathrm{Sign}(r),$$

$$g_r = g_q 1_{|r| \leq 1}.$$

{1.2. Backward propagation:}
{Note that the gradients are not binary.}
Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L and a^*
for $k = L$ to 1 **do**
  if $k < L$ **then**
    $g_{a_k} \leftarrow g_{a_k^b} \circ 1_{|a_k| \leq 1}$
  end if
  $(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$
  $g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$
  $g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$
end for

Пропагација градијената

Израчунавање и акумулација градијената

- Реалне вредности тежина су потребне да би стохастички градијентни спуст (SGD) уопште радио.
- SGD испитује простор у малим корацима са доста шума, међутим тај шум је упросечен доприносима градијената нагомиланим у свим тежинама.

Израчунавање и акумулација градијената

- Реалне вредности тежина су потребне да би стохастички градијентни спуст (SGD) уопште радио.
- SGD испитује простор у малим корацима са доста шума, међутим тај шум је упросечен доприносима градијената нагомиланим у свим тежинама.
- Овде је већа прецизност ипак неопходна

{2. Accumulating the parameter gradients:}
for $k = 1$ to L **do**
  $\theta_k^{t+1} \leftarrow \text{Update}(\theta_k, \eta, g_{\theta_k})$
  $W_k^{t+1} \leftarrow \text{Clip}(\text{Update}(W_k, \gamma_k \eta, g_{W_k^b}), -1, 1)$
  $\eta^{t+1} \leftarrow \lambda \eta$
end for

Shift-based Batch normalization

- BN се користи како би се убрзао тренинг и смањио утицај скале тежина.

Shift-based Batch normalization

- BN се користи како би се убрзао тренинг и смањио утицај скале тежина.
- Internal covariate shift.

Shift-based Batch normalization

- BN се користи како би се убрзао тренинг и смањио утицај скале тежина.
- Internal covariate shift.
- Множења заменити битовским померањем.

$$x \times y \rightarrow x \ll\gg \mathrm{AP2}(y)$$

Shift-based Batch normalization

- BN се користи како би се убрзао тренинг и смањио утицај скале тежина.
- Internal covariate shift.
- Множења заменити битовским померањем.

$$x \times y \rightarrow x \ll\gg \mathrm{AP2}(y)$$

- Једина операција која није битовско померање је инверзни корен, али постоје технике за његово апроксимирање.

Shift-based Batch normalization

- BN се користи како би се убрзао тренинг и смањио утицај скале тежина.
- Internal covariate shift.
- Множења заменити битовским померањем.

$$x \times y \rightarrow x \ll\gg \mathrm{AP2}(y)$$

- Једина операција која није битовско померање је инверзни корен, али постоје технике за његово апроксимирање.
- Није примћен губитак тачности приликом коришћења Shift-based BN уместо обичног BN.

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \ \{\text{mini-batch mean}\}$$
$$C(x_i) \leftarrow (x_i - \mu_B) \ \{\text{centered input}\}$$
$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(C(x_i) \lll\ggg \text{AP2}(C(x_i))) \ \{\text{apx variance}\}$$
$$\hat{x}_i \leftarrow C(x_i) \lll\ggg \text{AP2}((\sqrt{\sigma_B^2 + \epsilon})^{-1}) \ \{\text{normalize}\}$$
$$y_i \leftarrow \text{AP2}(\gamma) \lll\ggg \hat{x}_i \ \{\text{scale and shift}\}$$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i)$$

Batch normalization

Shift-based AdaMax

- ADAM такође смањује утицај скале тежина, собизором на велики број множења користи се Shift-based AdaMax.
- У експериметнима није примећен губитак тачности када се користи Shift-based AdaMax уместо обичног ADAM.

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \text{ (Update biased first moment estimate)}$$
$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \text{ (Update biased second raw moment estimate)}$$
$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) \text{ (Update parameters)}$$

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$
$$v_t \leftarrow \max(\beta_2 \cdot v_{t-1}, |g_t|)$$
$$\{\text{Updated parameters:}\}$$
$$\theta_t \leftarrow \theta_{t-1} - (\alpha \ll\gg (1 - \beta_1)) \cdot \hat{m} \ll\gg v_t^{-1})$$

Скаларни производ помоћу XNOR и popcount

A = 1 1 0 0 1

B = 0 1 0 0 1

(-1 * 1) + (1 * 1) + (-1 * -1) + (-1 * -1) + (1 * 1) = 3

Скаларни производ помоћу XNOR и popcount

A = 1 1 0 0 1

B = 0 1 0 0 1

(-1 * 1) + (1 * 1) + (-1 * -1) + (-1 * -1) + (1 * 1) = 3

XNOR(A, B) = 0 1 1 1 1

XNOR : (x == y) ? 1 : 0

p = popcount(XNOR(A, B)) = 4

N = 5

2*p – N = 3

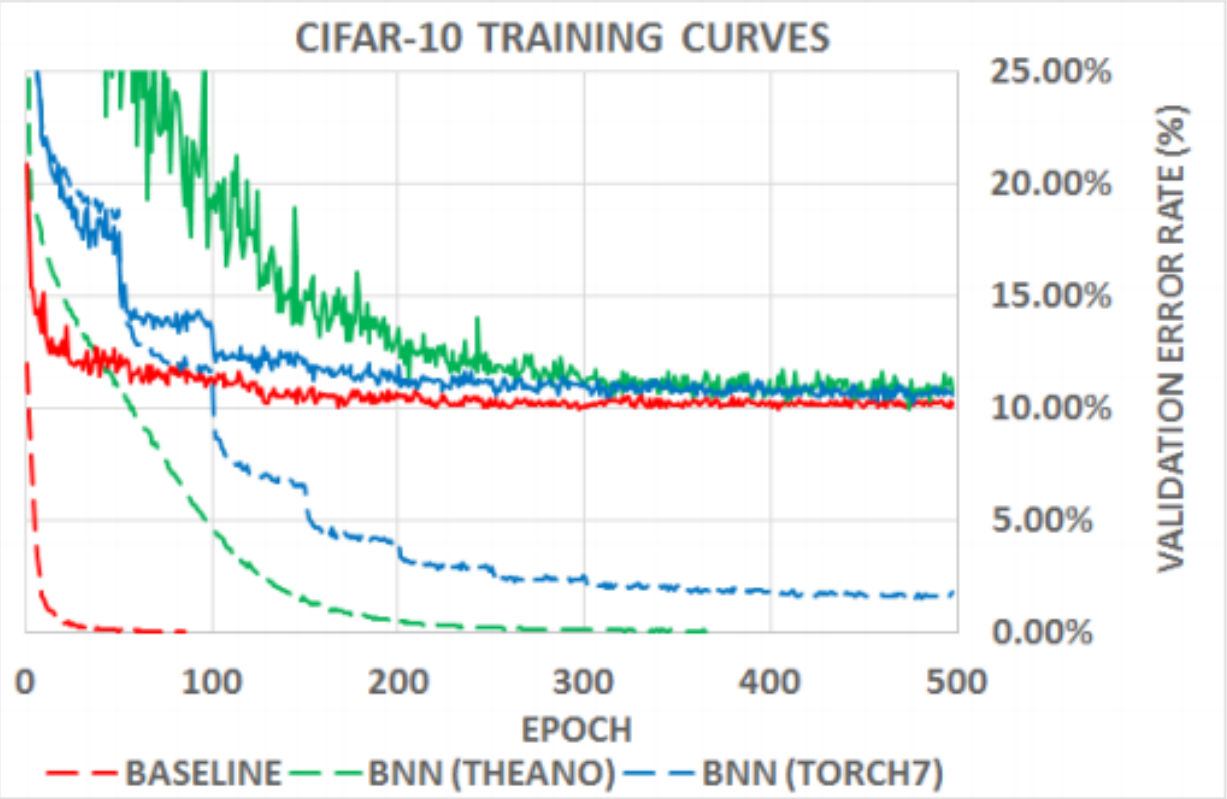
Резултати

Table 1: Classification test error rates of DNNs trained on MNIST (fully connected architecture), CIFAR-10 and SVHN (convnet). No unsupervised pre-training or data augmentation was used.

Data set	MNIST	SVHN	CIFAR-10
Binarized activations+weights, during training and test			
BNN (Torch7)	1.40%	2.53%	10.15%
BNN (Theano)	0.96%	2.80%	11.40%
Committee Machines' Array Baldassi et al. (2015)	1.35%	-	-
Binarized weights, during training and test			
BinaryConnect Courbariaux et al. (2015a)	1.29± 0.08%	2.30%	9.90%
Binarized activations+weights, during test			
EBP Cheng et al. (2015)	2.2± 0.1%	-	-
Bitwise DNNs Kim and Smaragdis (2016)	1.33%	-	-
Ternary weights, binary activations, during test			
Hwang and Sung (2014)	1.45%	-	-
No binarization (standard results)			
No reg	1.3± 0.2%	2.44%	10.94%
Maxout Networks Goodfellow et al. (2013b)	0.94%	2.47%	11.68%
Gated pooling Lee et al. (2015)	-	1.69%	7.62%

Резултати

Figure 1: Training curves for different methods on the CIFAR-10 dataset. The dotted lines represent the training costs (square hinge losses) and the continuous lines the corresponding validation error rates. Although BNNs are slower to train, they are nearly as accurate as 32-bit float DNNs.



Резултати

Table 3: Classification test error rates of the GoogleNet model trained on the ImageNet 1000 classification task. No unsupervised pre-training or data augmentation was used.

Model	Top-1	Top-5
Binarized activations+weights, during training and test		
BNN	47.1%	69.1%
Quantize weights and activations during training and test		
QNN 4-bit	66.5%	83.4%
Quantize activation,weights and gradients during training and test		
QNN 6-bit	66.4%	83.1%
No Quantization (standard results)		
GoogleNet - our implementation	71.6%	91.2%

Резултати

Table 5: Energy consumption of multiply- accumulations; see Horowitz (2014)

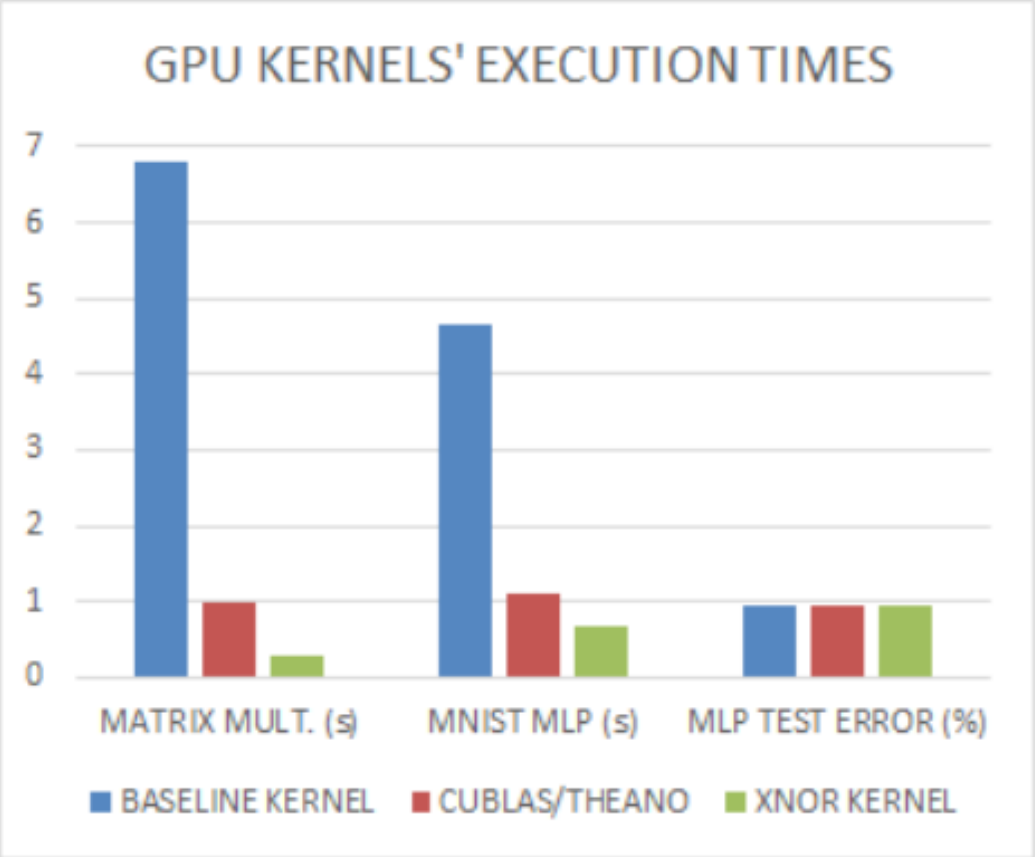
Operation	MUL	ADD
8-bit Integer	0.2pJ	0.03pJ
32-bit Integer	3.1pJ	0.1pJ
16-bit Floating Point	1.1pJ	0.4pJ
32-bit Floating Point	3.7pJ	0.9pJ

Table 6: Energy consumption of memory accesses; see Horowitz (2014)

Memory size	64-bit Cache
8K	10pJ
32K	20pJ
1M	100pJ
DRAM	1.3-2.6nJ

Резултати

Figure 3: The first 3 columns show the time it takes to perform a $8192 \times 8192 \times 8192$ (binary) matrix multiplication on a GTX750 Nvidia GPU, depending on which kernel is used. The next three columns show the time it takes to run the MLP from Section 3 on the full MNIST test set. The last three columns show that the MLP accuracy does not depend on the kernel



Хвала!