Guillaume Hannotte & Eugène Pascal Yaro

TP1&2 Application embarquée & Cross Compilation

<u>Objectif</u> : Le but des deux premiers TP est de mettre en place un CNN que vous allez choisir et faire l'apprentissage sur une base de données. La première base de données est MNIST.

Déroulement général :
1) Etudier les CNN déjà mis en place et en choisir un.
2) Compléter la couche de convolution (forward et backward) (fichier Layer.cu)
3) Mettre en place le réseau à partir de l'exemple de réseau mis à disposition. (fichier train.cpp)
4) Faire l'apprentissage de la base MNIST

Déroulement détaillé et questions :

Base de données

---

o Que contient la base de données MNIST ?
C'est une base de données de chiffres écrite à la main.

o Combien y a-t-il d'exemples dans la base de données de test / train de MNIST ?
Il y'a 10000 images de test et 60000 images d'apprentissage issues d'une base de données antérieures appelée NIST.

o Quelle est la taille des images ?
On retrouve 28 pixels de côté(28 × 28), soit 784 pixels

o A quoi correspondent les labels/classes ?
Les labels correspondent au chiffre affiché sur l'image et les classes correspondent aux chiffres

---

Choix CNN

---

o Reproduisez ou créez une architecture de CNN de classification simple (avec peu de couches)
décrivez la en spécifiant paramètres de chaque couches ainsi que les dimensions des données en entrée et en sortie de chaque couche utilisée au format (n,c,h,w) avec :

n la taille du batch
c le nombre de channels
h la hauteur
w la largeur.

Pour la taille du batch on a choisi **256** images d'apprentissage. L'objectif est d'avoir un batch avec le plus de données possibles et éviter un risque de dépasser la mémoire.

Voici les couches que nous avons sélectionnés:

couche de convolution

Dans le contexte d'un CNN la couche de convolution est impérative. C'est cette couche qui permettra d'extraites les caractéristiques.

Entrée: (1000, 64, 28,28)
Un choix de 64 filtres permet au modèle de rechercher un nombre satisfaisant de caractéristiques. Il sera plus performant. Plus de 64 risque de rendre le modèle trop complexe pour des données 28x28.
Pour les mêmes raisons que pour les filtres la dimension 5x5 de la matrice de convulsion permettra d'être précis et correctement identifier les classes tout en étant précis.

Sortie: (1000, 64, 24, 24). La sortie d'une convolution d'un matrice 28x28 par une matrice 5x5 donne bien une hauteur et largeur de 24.

Une couche d'activation

Nous avons choisi une couche d'activation de type Relu. Cette couche fixera les valeurs négative à 0. Cela permettra de rendre plus performant le réseau en généralisant plus facilement les données.

Cette couche est moins indispensable que des couches impératives. Cependant elle ajoute une optimisation qui augmentera le gain en performance de réseau.

une couche max pooling

Grâce à cette couche, on pourra réduire la taille des données des couches précédentes. Cela rendra le réseau plus robuste en réduisant les paramètres d'entrées.
On a choisi une hauteur et une largeur de 2x2.
Entrée: (1000, 64, 24, 24).
Sortie: (1000, 64, 12, 12). La matrice 2x2 de la couche max pooling à donc divisé la hauteur et la largeur de 2.

couche fully connected

Cette couche est donc la seconde plus importante avec la couche de convolution. Elle pourra déduire les classes en fonction du résultat des couches précédentes. Elle lie les caractéristiques précédemment identifiées par les autres couches pour obtenir la déduction.

Sachant que la base permet d'identifier 10 label pour les 10 chiffres, il a donc 10 classes. Ainsi on indiquera à cette couche un nombre de classe de 10.

Entrée: (1000, 64, 12, 12)
Sortie:(x1 x2…x10) Un vecteur de 1 dimension avec des valeurs pour les 10 classes.

Une couche Softmax.

C'est la 3 couche indispensable. Elle permet de récupérer des probabilités selon la couche précédente avec une fonction d'activation.

Entrée: (x1 x2…x10)
Sortie:(q1 q2…q10) Selon le paramètre Top 1 ou Top 5, on choisira soit la probabilité la plus élevé avec Top 1 pour faire la prédiction, soit l'une des 5 plus grandes valeurs avec le Top 5

Cela nous permettrait d'avoir de bons résultats de classification dans la base de données MNIST. Avec les couches de convolutions/fully connected/Softmax on pourra effectuer la classification. Les couches d'activations et Max Pooling permettent au réseau d'être plus performant.

CuDNN et mise en place CNN

Ainsi on obtient cela dans notre code:

```
Network model;

model.add_layer( new Conv2D("conv",64,5,5));
model.add_layer(new Activation("activ", CUDNN_ACTIVATION_RELU));
model.add_layer(new Pooling("pull",2,0,2,CUDNN_POOLING_MAX));
model.add_layer(new Dense("full",10));
model.add_layer(new Softmax("soft"));
model.cuda();
```

Question CuDNN et mise en place CNN

---

o A l'aide de la documentation CuDNN expliquez les différentes entrées des fonctions cudnnActivationForward et cudnnActivationBackward

La fonction cudnnActivationForward permet d'effectuer le calcul de l'activation d'une couche de neurones dans un réseau de neurones en utilisant un GPU.

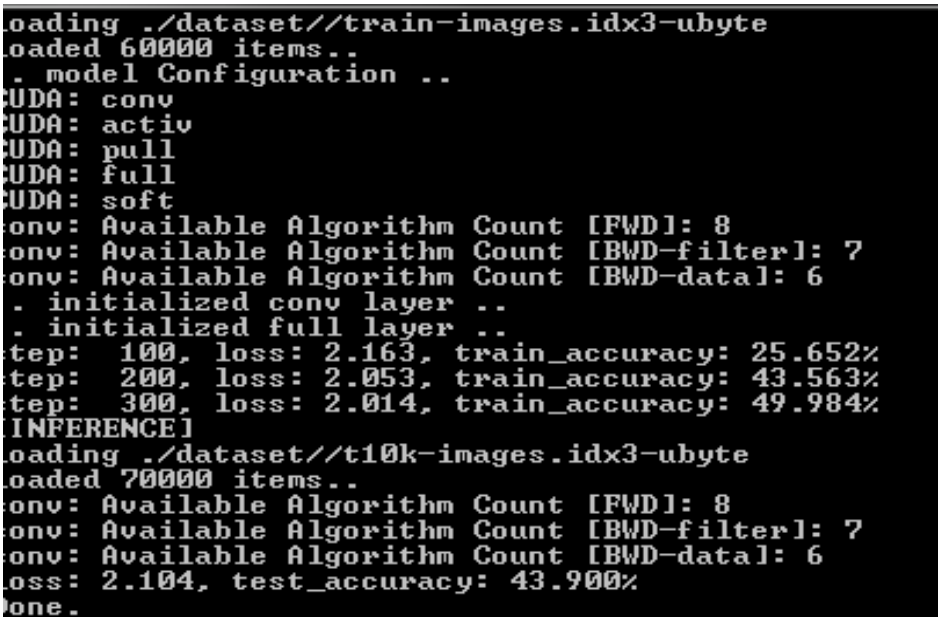
La fonction cudnnActivationBackward permet de calculer la dérivée de l'activation d'une couche de neurones dans un réseau de neurones en utilisant un GPU.

Le cudnnActivationForward est donc le signal envoyé d'une couche à une autre. Le cudnnActivationBackward est un signal envoyé dans les couches précédentes, il permet aux couches de connaître la modification des paramètres par rapport à la lost cost.

o Donnez les résultats de l'apprentissage (évolution de la précision et de la loss) ainsi que la précision obtenue sur les données de test.

model.add_layer(new Conv2D("conv",64,5,5));

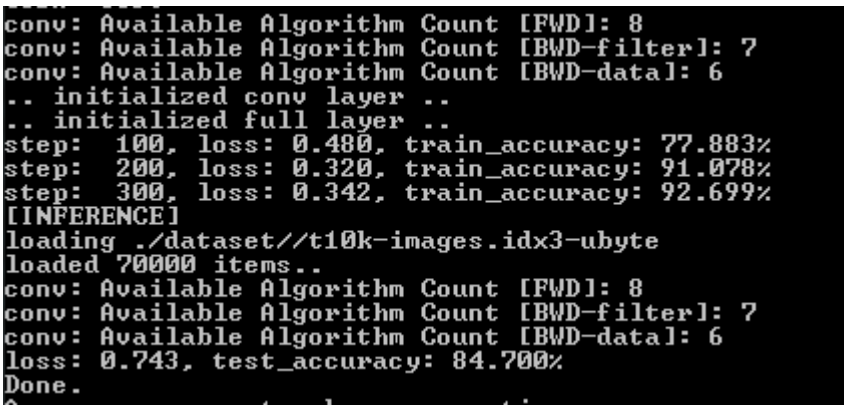
Nous avons choisi au départ pour la couche convolution un filtre avec un décalage égal à 5. Ce qui nous donne un train accuracy de 49.984 % et un test d'accuracy de 43.9 %.



o Modifiez les paramètres d'apprentissage pour améliorer l'entrainement du réseau

Pour l'amélioration de l'entraînement du réseau nous avons changé la valeur du décalage du filtre, nous sommes passés de 5 à 1.
Ce qui nous a permis d'améliorer le train accuracy égal à 92.622% et le test accuracy égal à 84.7%

```
model.add_layer( new Conv2D("conv",64,5,1));
model.add_layer(new Activation("activ", CUDNN_ACTIVATION_RELU));
model.add_layer(new Pooling("pull",2,0,2,CUDNN_POOLING_MAX));
model.add_layer(new Dense("full",10));
model.add_layer(new Softmax("soft"));
model.cuda();
```



Par ailleurs, nous avons essayé d'améliorer le test-accuracy mais nous avions une erreur lors de l'exécution du débogueur :
LNK1168 : impossible d'ouvrir CuDNN TP à compléter /x64/Debug/cudnn_project.exe pour écrire.

Analyse

---

o Y a-t-il une différence entre la précision sur les données d'entraînement vs données de test ?

Oui dans le dernier exemple on avait 92% de précisions avec les données d'entraînement et 84.7% avec les données de test

o Si oui, qu'est-ce que cela veut dire ?

Le modèle à surrappris par rapport aux données d'entraînement. A cause de cela, le modèle est moins capable de généraliser. Ainsi il est moins précis sur les données de test

(suite page suivante)

Guillaume Hannotte & Eugène Pascal Yaro

o Réalisez une matrice de confusion multi-classe
Matrice multi classe:

	Actual class									
Predicted class		0	1	2	3	4	5	6	7	8	9
	0	860	0	10	10	0	30	20	20	0	30
	1	0	1060	50	10	10	10	20	40	50	0
	2	10	10	810	20	20	10	30	20	20	10
	3	0	20	20	790	0	70	0	0	20	0
	4	0	0	20	0	920	60	20	30	10	40
	5	60	0	0	10	0	670	10	0	30	10
	6	20	0	10	10	10	30	840	0	10	0
	7	0	10	40	10	0	10	0	980	0	20
	8	20	60	30	30	0	20	0	30	700	50
	9	0	0	0	40	90	10	0	50	30	840

Matrice de confusion par classe (pour les autres questions):

		Label	
		0	**Non-0**
Output	**0**	860	120
	non-0	110	

F1 score= 0.88

		Label	
		1	**Non-1**
Output	**1**	1060	190
	non-1	100	

F1 score= 0.87

Guillaume Hannotte & Eugène Pascal Yaro

		Label	
		2	**Non-2**
Output	**2**	810	150
	non-2	180	

F1 score= 0.83

		Label	
		3	**Non-3**
Output	**3**	790	130
	non-3	140	

F1 score= 0.85

		Label	
		4	**Non-4**
Output	**4**	920	180
	non-4	130	

F1 score= 0.86

		Label	
		5	**Non-5**
Output	**5**	670	120
	non-5	250	

F1 score= 0.78

Guillaume Hannotte & Eugène Pascal Yaro

		Label	
		6	**Non-6**
Output	**6**	840	90
	non-6	100	

F1 score= 0.89

		Label	
		7	**Non-7**
Output	**7**	980	90
	non 7	190	

F1 score= 0.88

		Label	
		8	**Non-8**
Output	**8**	700	240
	non 8	170	

F1 score= 0.77

		Label	
		9	**Non-9**
Output	**9**	840	220
	non 9	160	

F1 score= 0.81

o Quelles sont les classes qui se confondent ?

Si on se réfère au F1 score, le 8 et 5 sont le plus souvent confondus

Guillaume Hannotte & Eugène Pascal Yaro

o Quelles sont les classes bien discriminées ?

Si on se réfère au F1 score, le 7, 6 et 0 semble être bien discriminé

o Qu'est-ce que vous en déduisez ?

Le 2 et le 5 se ressemblent ce qui peut expliquer la confusion. Le 8 également peut étre confondu avec un 0.

Le 7 et le 6 semblent avoir une forme originale; ainsi ils sont moins confondus.

En résumé la confusion dépend de la forme original d'un chifrre