TP4 Byte-Sized Brains : The Pocket AI Lab

Auteurs : Cédric Rosat, Costantino Cecchet

Weights quantization

Record the range of the weights, as well as their 3-sigma range (the difference between μ−3σ and μ+3σ).

```
Weights values ranges:*
    - conv1 values range: -0.5752065181732178 - 0.5305084586143494
    - conv2 values range: -0.5465968251228333 - 0.4471343457698822
    - fc1 values range: -0.2240881621837616 - 0.26401472091674805
    - fc2 values range: -0.2140781581401825 - 0.27434059977531433
    - fc3 values range: -0.5074619650840759 - 0.39367854595184326
```

```
Weights sigma ranges:*
    - conv1 3-sigma range: -0.5953277945518494 - 0.5816145539283752
    - conv2 3-sigma range: -0.35913988947868347 - 0.334823876619339
    - fc1 3-sigma range: -0.13010206818580627 - 0.1253042221069336
    - fc2 3-sigma range: -0.19376078248023987 - 0.18946179747581482
    - fc3 3-sigma range: -0.3818199634552002 - 0.3778454065322876
```

*Ces valeurs peuvent être différentes de celles dans le notebook.

Explain which range you used for your quantization. Does range have an impact on model performance in this case ? Explain your answer.

Commençons par analyser les performances sur le jeu de test avec aucune quantization, puis de quantization de 1, 2 et 3 écarts-types autour de la moyenne (afin d'avoir des valeurs représentatives) :

```
Accuracy of the network on the test images with no quantization: 53.41%*
Accuracy of the network on the test images with 3-sigma quantization:
52.67%*
Accuracy of the network on the test images with 2-sigma quantization:
51.33%*
Accuracy of the network on the test images with 1-sigma quantization:
41.4%*
```

*Ces valeurs peuvent être différentes de celles dans le notebook.

Réduire la largeur de la plage permet de gagner en espace mémoire et en vitesse d'inférence, mais impacte négativement la précision du modèle. En s'appuyant sur les valeurs ci-dessus, on peut conclure qu'une

plage coupée à 2 écarts-types est un bon compromis entre réduction de la taille et préservation de la précision.

Do you observe a drop in the general accuracy ? If you did everything right, it should be negligible. Explain your findings.

Comme expliqué ci-dessus, trop rapprocher les valeurs extêmes autour de la moyenne impacte négativement les données. Voici le nombre de valeurs qui sont hors de la plage en fonction du nombre d'écarts-types :

```
Percentage of weights outside n-sigma range*
- conv1
    - 1-sigma range: 33.3%
    - 2-sigma range: 4.89%
    - 3-sigma range: 0.0%
- conv2
    - 1-sigma range: 28.5%
    - 2-sigma range: 5.12%
    - 3-sigma range: 0.667%
- fc1
    - 1-sigma range: 30.6%
    - 2-sigma range: 4.55%
    - 3-sigma range: 0.523%
- fc2
    - 1-sigma range: 36.0%
    - 2-sigma range: 2.95%
    - 3-sigma range: 0.169%
- fc3
    - 1-sigma range: 29.0%
    - 2-sigma range: 5.36%
    - 3-sigma range: 0.595%
```

*Ces valeurs peuvent être différentes de celles dans le notebook.

On remarque qu'avec 3-écarts-types, entre 28% et 34% des valeurs sont "perdues" car limitée au maximum et au minimum de la plage. Cette perte se fait ressentir sur le modèle qui perd 12% de précision. Alors qu'avec une quantization à 2 écarts-types, la perte en précision n'est que de 2,1%.

Compare the memory footprint of the original model and the quantized one. Did the memory footprint change ? Explain your findings. You can use torchinfo or torch-summary to get the memory footprint.

Voici la taille en mémoire des divers poids des modèles quantizé ou non :

```
Taille en mémoire du modèle de base: 0.2356 Mo
Taille en mémoire du modèle quantifié avec 3-sigma: 0.2356 Mo
Taille en mémoire du modèle quantifié avec 2-sigma: 0.2356 Mo
Taille en mémoire du modèle quantifié avec 1-sigma: 0.2356 Mo
```

La taille des poids ne change pas. Même si les valeurs sont des entiers entre -128 et 127, leur représentation en mémoire reste un `float` et aucun gain de mémoire, ni de temps d'inférence, n'est fait. Stocker les poids sur un entier 8 bits permettrait d'obtenir un gain en mémoire.

Activation quantization

Record the range of the activations, as well as their 3-sigma range.

```
Activations values ranges:*
    - conv1 values range: -1.0 - 1.0
    - conv2 values range: 0.0 - 8.476506233215332
    - fc1 values range: 0.0 - 15.80102252960205
    - fc2 values range: 0.0 - 11.326419830322266
    - fc3 values range: 0.0 - 7.495452404022217
```

```
Activations sigma ranges:*
    - input 3-sigma range: -1.5791903278671173 - 1.5015859032489558
    - conv1 3-sigma range: -1.7721760781304239 - 2.921475828344227
    - conv2 3-sigma range: -2.7820793193116806 - 4.109955156077108
    - fc1 3-sigma range: -2.369422747677803 - 3.267471729166893
    - fc2 3-sigma range: -1.5978766447063344 - 2.2510592511130616
    - fc3 3-sigma range: -7.273087554955707 - 6.98465022621726
```

*Ces valeurs peuvent être différentes de celles dans le notebook.

Develop a formula for the quantized output of conv1 : O1q, as a function of sW 1, sO1, sI , Iq, W1q.

En se basant sur les équations suivantes : $$ O = W*I $$

$$ W \approx s_W * W_q $$

On peut écire l'équation pour quantizer une couche : $$ s_{O1} * O_{1q} = s_{W1} * W_{1q} * s_I * I_q $$
Develop a general formula for the quantized output of convi : Oiq, as a function of sW i, sOi, sI , Iq, Wiq, and the scaling factors of the previous layers. Try to do it recursively by starting with conv2q to extract the formula.

L'équation (3) ne prend en compte qu'une seule couche. Afin d'avoir une équation pour quantizer tout notre modèle, nous devons prendre en compte toutes les échelles *s* afin d'obtenir l'échelle totale du modèle (*scale* dans l'équation (4)) : $$ O_q = scale * \prod{} W $$ En s'aidant de l'équation (3), on peut chaîner cette équation pour chaque couche de notre modèle, et obtenir l'équation générale du modèle : $$ O_{qn} = \prod_{i=1}^{n} S_{Wi} * W_{iq} * \prod_{i=1}^{n-1} S_{Oi} * I * S_I \space \space \space \text{avec n le nombre de couches} $$ Où notre scale finale est : $$ scale = \prod_{i=1}^{n} S_{Oi} * S_I \space \space \space \text{avec n le nombre de "couches" de connections entre couches (nombre de couches - 1)} $$

NNTool quantization

What's the objective of the adjust command ?

La commande `adjust` dans NNTool est utilisée pour préparer le modèle pour l'exécution sur des kernels AT. Cela inclut l'ajout et la suppression de transpositions, la mise à jour des dimensions du graphe pour assurer que toutes les couches soient cohérentes.

With the help of the nnTool documentation, provide the right command for layer fusions.

```
fusions --scale8
```

On applique le paramètre `--scale8` afin d'utiliser une échelle 8 bits (notre échelle va de -128 à 127, soit 8 bits).

Which layers have been fused ?

=> Première couche convolutive :

- Avant Fusion :
  - Convolution : `_conv1_Conv`
  - Pooling : `_pool_MaxPool`
  - ReLU : `_Relu`
- **Après Fusion** : `_conv1_Conv_fusion`

=> Deuxième couche convolutive :

- Avant Fusion :
  - Convolution : `_conv2_Conv`
  - Pooling : `_pool_1_MaxPool`
  - ReLU : `_Relu_1`
- **Après Fusion** : `_conv2_Conv_fusion`

=> Première couche "fully connected" :

- Avant Fusion :
  - MatMul : `_fc1_MatMul`
  - ReLU : `_Relu_2`
- **Après Fusion** : `_fc1_MatMul_fusion`

=> Deuxième couche "fully connected" :

- Avant Fusion :
  - MatMul : `_fc2_MatMul`
  - ReLU : `_Relu_3`
- **Après Fusion** : `_fc2_MatMul_fusion`

Based on the first parts of this lab, explain why we need a set of images for our quantization.

Avant de pouvoir quantizer un modèle, il faut l'entraîner afin de mettre à jour les poids et les couches d'activation. Une fois que le modèle est entraîné et qu'il a apprit les caractéristiques des images, on peut le quantizer son "savoir" pour réduire la représentation en mémoire.

Comme la quantization engendre une perte d'informations, il faut dans un premier temps avoir ces informations.

What should be the properties of this set of images ? Think in terms of diversity of images.

Ces images doivent être variées. Les diverses classes doivent avoir suffisamment d'images et le même nombre d'images par classe. Chaque classe doit elle même être suffisamment diversifiée avec différentes prises de vues, différentes conditions d'éclairage etc. De plus, les images doivent correspondre à la taille attendue en entrée du réseau.

In your generated file model.h, find the quantization constants (OUT_SCALE) and explain why they may be different than the ones you had to compute in stages 1 and 2.

Pour calculer les quantizations dans le fichier `model.h`, on a dû fournir une image spécifique afin de créer des tenseurs qui ont ensuite été utilisé dans la commande suivante : `nntool -g -M . -m model.c -T tensors -H model.h model_lab4.json`. Alors que dans notre notebook Jupyter, la quantization a été réalisée sur un modèle entraîné (qui a vu toutes les images du jeu d'entraînement). Cette différence explique la différence entre les échelles de quantization.

S'ajoute à cela les méthodes de calcul différentes. Dans notre notebook, on a utilisé l'approche des écarts-types, alors que dans NNTool, `--scheme SQ8 a été utilisé`.

Et pour finir, NNTool effectue des optimisation supplémentaires après le calcul initial des échelles pour améliorer la performance ou réduire l'erreur de quantization.