Opgaver – Uge 6

Opgave 2: Færdig

1. **Name five areas where deep learning is now the best in the world.**
   * Computer vision,
   * Image generation
   * Playing games
   * Recommendation system
   * Medicine
2. **Based on the book of the same name, what are the requirements for parallel distributed processing (PDP)?**
   * Et set af proces enheder
   * En tilstand af aktivering
   * En outputfunktion for hver enhed
   * Et mønster til tilslutning ” pattern of connectivity” mellem enheder
   * En formerings regel til udbredelse af mønstre for aktiviteter gennem netværket af tilslutningsmuligheder
   * En aktiveringsregel for at kombinerer de input som har en betydning for enheden med dens nuværende tilstand til at fremstille et output for enheden.
   * En læringsregel hvor ”pattern of connectivity” bliver ændret af erfaring
   * et miljø hvor systemet kan arbejde.
3. **What were the two theoretical misunderstandings that held back the field of neural networks?**

De antog at det ikke var muligt at bruge neural networks. De antog at kun et lager mere (altså 3) var nok til at lave et neural netværk

1. **What is a GPU?**

Det er special slags processor i en computer som kan håndtere tusindvis af opgaver påamme tid.

1. **Why is it hard to use a traditional computer program to recognize images in a photo?**

Traditionelle computere programmer er ikke bygget til at lærer, de er hardcodet til at lave en bestemt løsning frem for at lære. Hvor at aflæsning af billeder næsten er umulig at hardcode

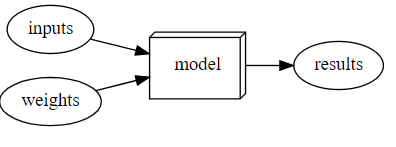
1. **What did Samuel mean by "weight assignment"?**

Er et særlig valg af værdier til de værdier som går ind og definerer hvordan programmet skal fungere

1. **What term do we normally use in deep learning for what Samuel called "weights"?**

Modelparameter

1. **Draw a picture that summarizes Samuel's view of a machine learning model.**



1. **Why is it hard to understand why a deep learning model makes a particular prediction?**

Den fungere meget som en blackboks, vi giver den noget data og får et resultat.

1. **What is the name of the theorem that shows that a neural network can solve any mathematical problem to any level of accuracy?**

Universal approximation theorem

1. **What do you need in order to train a model?**

Data og labels

1. **How could a feedback loop impact the rollout of a predictive policing model?**

Hvis modellens resultat vil medføre i yderligere inputs, kan denne input have en manipulations effekt på modellens kommende resultater som vil medføre til yderligere output af samme type og format. Dette giver en gentagende effekt og misviser virkeligheden.

1. **Do we always have to use 224×224-pixel images with the cat recognition model?**

Man skulle i starten, men nu skal man ikke. Jo større billedet er jo bedre vil modellen virke

1. **What is the difference between classification and regression?**
   * Classification modellen er en som prøver at forudsige en klasse eller kategori, hvor den forholder sig til en bestemt antal muligheder. Eksemplet er hund og kat modellen.
   * Regression modellen er en som prøver at forudsige en eller flere numeriske mængder, så som temperatur eller en lokation.

Opgave 3: Færdig

1. **What is a validation set? What is a test set? Why do we need them?**

**Validation set, validere modellens data.**

Test set, yderligere validere modellens data unden noget bios.

1. **What will fastai do if you don't provide a validation set?**

Opdele selv data’en i tre dele

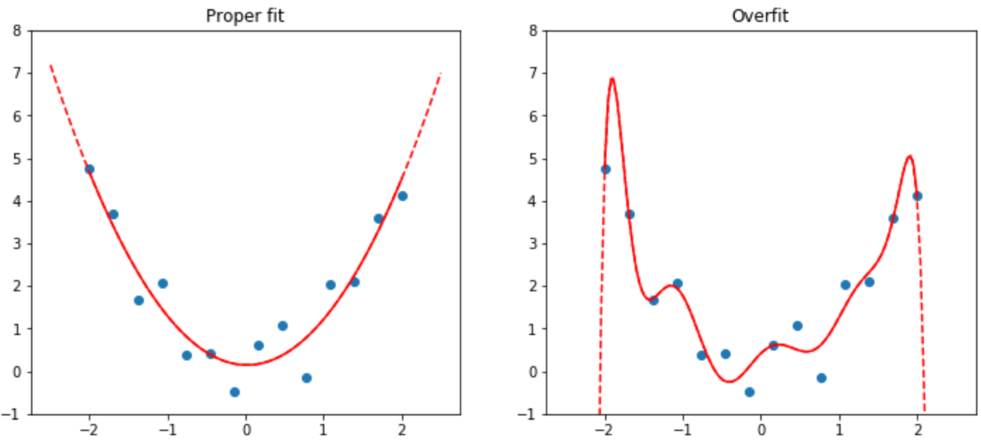
1. **Can we always use a random sample for a validation set? Why or why not?**

Hvis noget er tid estimeret, kan du ikke hente data’en randomt

1. **What is overfitting? Provide an example.**

Det er når modellen, i stedet for at finde mønster begynder at huske data’en i stedet for.

Eksempel:



1. **What is a metric? How does it differ from "loss"?**

Metric er en funktion der måler kvaliteten af modellens forudsigelser ved brug af validation sættet og vil blive printet ud i slutningen af hver Epochs.

Metric bliver aflæst af mennesker, så en god metric er en som er nem at forstå. Den minder meget om Loss, dog er den ikke det samme, siden loss definere ”measure of performance” som træningssystemet bruger til at opdatere weights automatisk.

1. **How can pretrained models help?**

I stedet for at lave en model fra starten, så kan man benytte sig af en pretrained model, som allerede har lært nogle af de mønstre som man ønsker.

1. **What is the "head" of a model?**

Det er hvad du tilføjer til den pretrained model således at den gør hvad du ønsker

1. **What kinds of features do the early layers of a CNN find? How about the later layers?**

Det første lag finder generelle ting, så som former og figur.

Det sidste lag finder koncepter.

Jo flere lag jo mere specifik.

1. **Are image models only useful for photos?**

Nej, den kan bruges til alt hvad der har et visuelt mønster.

1. **What is an "architecture"?**

Den matematiske funktion som vi putter input parameter i.

1. **What is segmentation?**

At skabe en model som kan genkende indholdet af hvert pixel i et billede bliver kaldt ”segmentation”

1. **What is y\_range used for? When do we need it?**

Det er brugerdefineret rækkevidde for hvilken rækkevidde den må gætte på.

1. **What are "hyperparameters"?**

Parameter som er fastsat af udvikleren, som ikke kan ændres af modellen

1. **What's the best way to avoid failures when using AI in an organization?**

Det er vigtigt at du benytter dig af dine egne test data, når du modtager en pretrained model. For at sikre at den ikke er overfittet.

1. **Complete the Jupyter Notebook online appendix. (use the file app\_jupyter.ipynb in Gradient or get it from github:** [**https://oreil.ly/9uPZe**](https://oreil.ly/9uPZe)**)**
2. **Why is a GPU useful for deep learning? How is a CPU different, and why is it less effective for deep learning?**

En GPU kan løse mange flere løsninger på samme tid end hvad den CPU kan.

1. **Try to think of three areas where feedback loops might impact the use of machine learning. See if you can find documented examples of that happening in practice.**

Politi eksemplet

Skak eksemplet

Uge 7

Opgave 1: Ikke færdig

Opsæt hver især en blog med fastpages på samme måde. Læg jeres besvarelser fra sidste gang ind på bloggen. Bloggen skal bruges som dokumentation til eksamen, og derfor vælger vi fastpages, for så kan vi lægge notebooks ind.

Opgave 2: Fejl ved udførelse

Lav samme løsning som bogen i kapitel 2, men i stedet for bjørne, skal I lave en Hotdog/Not Hotdog classifier, brug min kode og min api-nøgle til at hente billeder.

Problem: Opstår ved kald på results = search\_images\_bing(key, 'hotdog') årsagen til dette er at den nøgle vi har fået udleveret, er ikke længere gyldig, fordi licensen er udløbet.

Uge 8

Opgave 3:

1. **Explain how the "pixel similarity" approach to classifying digits works.**

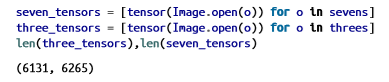
Der laves et tensor stack, hvor der tages et average af hvert pixel, som så sammenlignes med det ideelle billed af et 3- eller 7- tal

1. **What is a list comprehension?**

Syntax: list= [*expression* for item in iterable if condition]

En list comprehension er en kort måde at lave en liste på.

Eksempler:

  
Create one now that selects odd numbers from a list and doubles them.

double\_odd = [double\_numbers(i) for i in numbers if i%2 == 1]

1. **What is a "rank-3 tensor"?**

Det er en liste af rank-2 matrix

1. **What are RMSE and L1 norm?**

Root Mean Square Error : forskellene mellem værdier forudsagt af en model eller en estimator og de observerede værdier.

L1 norm : The L1 norm that is calculated as the sum of the absolute values of the vector.

1. **Create a 3×3 tensor or array containing the numbers from 1 to 9.   
   Double it.**

  
Select the bottom-right four numbers.

1. **What is broadcasting?**

Det er sort magi der udvider tensor med end mindre rank til at matche den med en stør rank.

1. **Are metrics generally calculated using the training set, or the validation set?**

Validation

1. **What is SGD?**

Stochastic Gradient Descent :

Det er en optimizer der kan håndtere stor datasets idet den samler det i mini-batches

1. **What are the seven steps in SGD for machine learning?**
   1. Init
   2. predict
   3. loss
   4. gradient
   5. step
   6. repeat?
   7. stop
2. **How do we initialize the weights in a model?**

vi tager random værdier for at starte med, da det alligevel bliver tilpasset

1. **What is "loss"?**

loss is a number indicating how bad the model's prediction was on a single epoch. If model prediction is perfect the loss is 0.

1. **Why can't we always use a high learning rate?**

Med en for stor learning rate, laver vi for store steps, hvormed vi ikke kan komme til bedste værdier. Den skyder over målet

1. **What is a "gradient"?**

En gradient er værdien på et fald eller stigning fra et punkt på en graf.

“Gradient is denied as rise/run; that is, the change in the value of the function, divided by the change in the value of the parameter.”

1. **Why can't we use accuracy as a loss function?**

accuracy er for mennesker

loss bliver brugt for modellen

1. **What is the difference between a loss function and a metric?**

loss er antallet af forkerte predictions, mens metric er optimeret for mennesker, det viser i % hvor højt scoren er i forhold til rigtige predictions

1. **What is the function to calculate new weights using a learning rate?**

w -= gradient(w) \* lr

1. **What does the backward method do?**

backwards methoden laver backpropagation, som optimizer weights, og dermed “træner” modellen

Opgave 6: fejl ved udførelse

Arbejd videre med ”*Who painted the painting?*” (enten i Gradient eller Colab).  
Eksperimenter f.eks. med at tune modellen f.eks. vha. data-cleaning eller en anden arkitektur (anvend f.eks. r*esnet30* eller *resnet50* i stedet for *resnet18*).

Problem: samme problem som i uge 7 opgave 2.

Uge 9

Opgave 1:

Opgaven er: I skal bygge jeres egen learner fra bunden af ved at lave en klasse I Python der hedder MyLearner eller noget i den stil. Learneren skal indeholde de 7 punkter bogen beskriver på side 150, og som gennemgås af to omgange fra hhv. side 159-162 og igen side 171-176.

Lad være med at google løsningen, men kig i bogen og gør som de gør.

Opsæt derefter et neuralt netværk som vist på side 178, men eksperimenter med at gøre netværket dybere og/eller med flere noder pr lag, og se, hvad der performer bedst, ved at udskrive accuracy pr epoch.

Test med 3-tal og 7-tal fra MNIST (eller med dit eget hotdog-datasæt).

Opgave 2:

Fordel jer igen/stadig I break-out rooms, og lav opgave 2 i bogen kapitel 4 under Further Research. Brug din egen learner til opgaven, og træn den til at genkende alle tallene I MNIST (eller alle kunstnerne fra dit Maleri-datasæt).

Uge 10

Opgave 4:

Vælg en af følgende opgaver (eller kombiner efter eget ønske):

* See if you can improve the accuracy of the classifier in this chapter. What is the best accuracy you can achieve?
* Opgaven ”Who painted the painting?” er en multi-kategori opgave (en kategori for hver maler du har billeder fra).   
  Brug den til at arbejde med de enkelte teknikker fra kapitlet (f.eks. DataBlock ,item resize, batch resize, RegexLabeller (måske skal du overveje navngivning af billedfilerne), summary, softmax, nll\_loss, CrossEntropyLoss, lr\_find, fine\_tune, fit\_one\_cycle, unfreeze, valg af antal epochs mm.)
* Hvis dine data til ”Who painted the painting?” ikke er velegnede kan du finde et velegnet multi-kategori dataset online.

Uge 11

Opgave 1:

Opgaven i dag kommer til at løbe over to uger. I denne uge skal I skal bruge det I har lært indtil nu til at løse opgaven fra zindi.africa (se nedenfor). Upload jeres løsningsark til sitet og få en score tilbage. I næste uge, skal I så forsøge at forbedre resultatet ved at bruge de optimeringstricks, der bliver gennemgået i næste kapitel.

Opgaven findes her:

<https://zindi.africa/competitions/iclr-workshop-challenge-1-cgiar-computer-vision-for-crop-disease>

Hvis datasættet driller, så hent det her:

<https://www.kaggle.com/shadabhussain/cgiar-computer-vision-for-crop-disease>

Her er den evaluation metric der bruges til at vurdere, hvor godt I har klaret opgaven:

LogLoss=−1n∑i=1n[yilog(ŷ i)+(1−yi)log(1−ŷ i)],

Findes den allerede i fastai? Hvis ikke kan I jo overveje, hvordan man implementerer den og får den vist per epoch. Og hvis den findes, kan I jo sætte den på.

Hvis I når at blive færdige med zindi.africa-opgaven, så kan I vælge mellem følgende to opgaver:

## Besvar en recaptcha med kode

Graphical user interface, application, Word

Description automatically generated

I skal lave en løsning, hvor I skriver captions til billeder. Fx: Biler, skorsten, en brandhane, et fordgængerfelt etc.

Bagefter kan I jo prøve, om I via javascript kan downloade billederne fra en recaptcha og identificere dem korrekt med jeres model, og igen via javascript kan udvælge de rigtige billeder og trykke på submitknappen . Forskellen mellem et valgt billede og et ikke-valgt er blot en ekstra css-klasse. Se test-recaptchas her:

<https://www.google.com/recaptcha/api2/demo>

Eller:

<https://patrickhlauke.github.io/recaptcha/>

## Kaggle competiton

Denne opgave er til jer, som gerne vil have lidt ekstra ved siden af SU’en. Opgaven bør kunne løses med det I har lært indtil nu, men første udfordring er faktisk at få udpakket zipfilen. Jeg vil anbefale at man udpakker dele af zipfilen fra flere forskellige tråde, men derfor kan det stadigvæk godt være det tager lang tid.

Konkurrencen er her:

<https://www.kaggle.com/c/bms-molecular-translation/overview>

Uge 12

Opgave 1

1. **What is the difference between ImageNet and Imagenette? When is it better to experiment on one versus the other?**

Imagenette er et datasæt lavet af 10 vidt forskellige klasser fra Imagenet. Imagenette er bedre at experimentere på når man vil træne hurtigere for at se hvad der fungere på det større også.

1. **What is normalization?**

At vores input data har et gennemsnit på 0 og en standardafvigelse på 1.

1. **Why didn't we have to care about normalization when using a pretrained model?**

Her brugte vi CNN\_learner som selv normaliserede vores data for os. Ellers kan det netop være vigtigt at vi normaliserer vores data på samme måde som er blevet gjort i forvejen.

1. **What is progressive resizing?**

Man starter med at træne sin model med små billeder, og ende med at træne med større billeder.

1. **What is test time augmentation? How do you use it in fastai?**

Ved skabelse af validerings sættet, kan man hakke hvert billede ud i flere dele og sende dem igennem modellen. Fastai vil som standard lave en center hakning af billedet, dette vil dog ofte medføre at kanterne i billedet ikke bliver taget med.

1. **Is using TTA at inference slower or faster than regular inference? Why?**

Det er langsommere fordi man skal inferere over flere augmentede billeder og finde gennemsnittet i stedet for at kun gøre det på et billede.

1. **What is Mixup? How do you use it in fastai?**

Det er et værktøj til at undgå overfitting.

Man kan bruge det ved at sætte den på cbs(callbacks) i constructoren for vores learner.

1. **Why does Mixup prevent the model from being too confident?**

Den lægger to billeder over hinanden som modellen så skal predicte hvilke er og hvad deres vægt er.

1. **Why does training with Mixup for five epochs end up worse than training without Mixup?**

Fordi Mixup laver et nyt Mixup hver eneste gang man kigger på et billede, så efter 5 epoker har den ikke nået at kigge nok på de samme ting til at kunne se et mønster. Den skal angiveligt træne i over 80 epoker for at være bedre.

1. **What is the idea behind label smoothing?**

Det er at undgå overfitting, hvor label smoothing går ind og gør modellen mindre sikker på sine resultater. Derved bliver modellen bedre til at generalisere.

1. **What problems in your data can label smoothing help with?**

Det er mere robust i tilfælde af fejllabeled data.

1. **When using label smoothing with five categories, what is the target associated with the index 1?**

[0.01, 0.96, 0.01, 0.01, 0.01] = 0.96

1. **What is the first step to take when you want to prototype quick experiments on a new dataset?**

Start med at lave et lille subset af dit dataset, så du hurtigt kan experimenter med det, i stedet for at bruge et kæmpe dataset, der vil tage alt for lang tid at træne.

Opgave 2

Fortsæt med *zindi.africa* opgaven fra sidste uge. Anvend emnerne fra denne uges pensum (normalization, progressive resizing, test time augmentation, mixup og label smoothing) og se om du får bedre resultater.

Uge 15:

Opgave 1

Med CL-strukturen tænk-par-del, skal I nu først, hver især, nedskrive jeres egen forståelse af følgende begreber. Til det får I 15 min.

Derefter sætter jeg jer sammen to og to, hvor I udveksler jeres forståelser, og bliver enige om en fælles forståelse af begreberne (I må gerne tage bogen eller andre ressourcer i brug til at hjælpe hukommelsen). Til dette er der 10 min.

Til sidst bliver I sat sammen parvis for endnu engang at tjekke jeres forståelser. Det får I 5 min til. På denne burde I få en vis mængde sikkerhed for, hvad begreberne dækker over. Der er selvfølgelig den mulighed at alle tager fejl, men der er stor sandsynlighed for, at I efter en snak vil være kommet frem til den korrekte forståelse fordi fire hoveder tænker bedre end ét J

**Begreberne er:**

Convolution

Kernel

Stride

Padding

Filter

Batchnorm

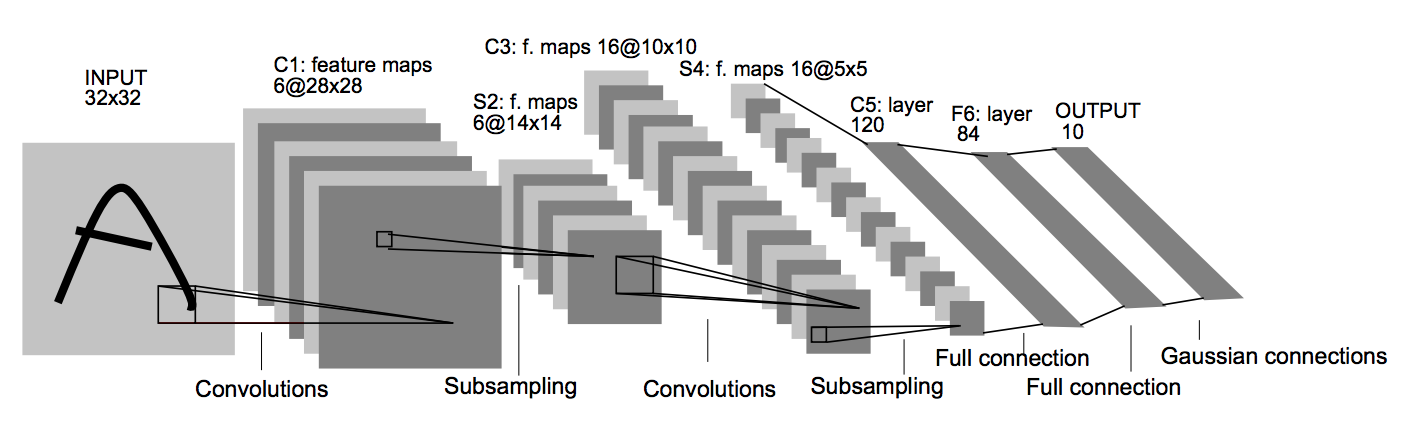
Opgave 3

Implementér op til tre af følgende arkitekturer, og sæt dem ind i en learner med et passende datasæt.

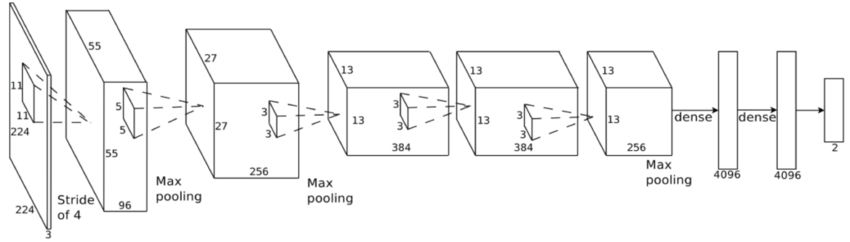
Første del af opgaven, er at forstå, hvordan man skal forstå billedet. Næste del at implementere det. Der bruges begreber, som I ikke har set før, så som Maxpool og Averagepool som vil blive delvist forklaret i næste kapitel af bogen (og i videoen ovenfor), men også ord som Full connection eller dense, som I måske har en idé om hvad betyder.

I må gerne læse den videnskabelige artikel bag arkitekturen, hvis det hjælper

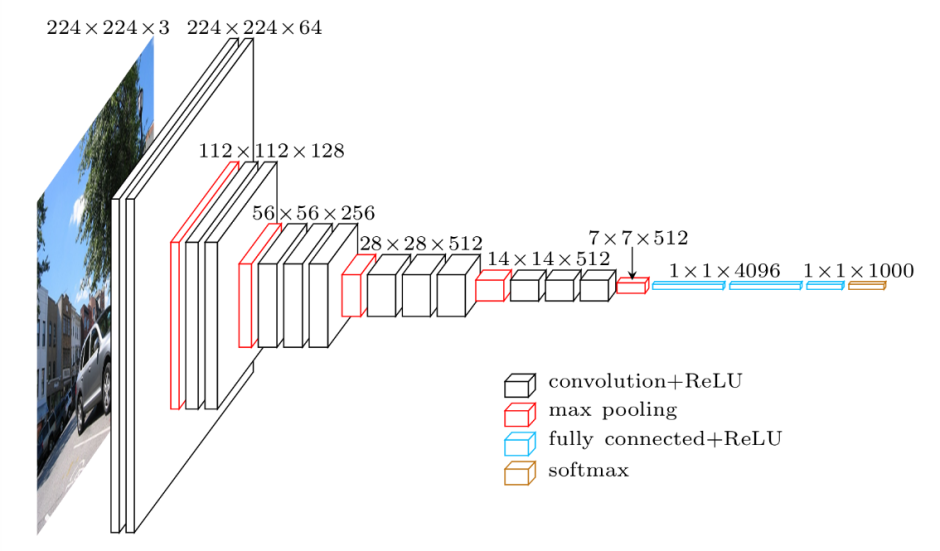
1. LeNet-5:  [Gradient-based learning applied to document recognition](http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf)



1. AlexNet: [ImageNet Classification with Deep Convolutional Neural Networks](https://www.nvidia.cn/content/tesla/pdf/machine-learning/imagenet-classification-with-deep-convolutional-nn.pdf)



1. VGG-16:  [Very Deep Convolutional Networks for Large-Scale Image Recognition](https://arxiv.org/abs/1409.1556)



Uge 16

Opgave 1

1. **What is a "skip connection"?**

værdierne af et output bliver til inputtet på flere lag, hvor der ikke bliver gjort noget ved disse, lag bliver altså “skippet”, når en gradient kommer for tæt på 0.

1. **Why do skip connections allow us to train deeper models?**

Det gør det muligt at undgå gradient degradation, således at der ikke kan være for mange lag i en model.

1. **What is "identity mapping"?**

Returnerer inputtet uden at ændre det

1. **What do ResNets have to do with residuals?**

ResNets prøver ikke at komme med et output fra et layer, men i stedet for med at minimere differencen mellem outputtet og det ønskede resultat. Dermed er ResNets god til at lære om der er små ændringer eller om der skal skippes.

“*If the outcome of a given layer is x and we’re using a ResNet block that returns y = x + block(x), we’re not asking the block to predict y; we are asking it to predict the difference between y and x. So the job of those blocks isn’t to predict certain fea‐ tures, but to minimize the error between x and the desired y. A ResNet is, therefore, good at learning about slight differences between doing nothing and passing through a block of two convolutional layers (with trainable weights). This is how these models got their name: they’re predicting residuals (reminder: “residual” is prediction minus target).*” - fra side 448

1. **How do we deal with the skip connection when there is a stride-2 convolution? How about when the number of filters changes?**

Ved at bruge *fully convolutional networks* hvor man tager gennemsnittet af activations henover et convolutional grid, dermed bliver et grid af flere activation til én activation pr. billede.

1. **What is the "stem" of a CNN?**

Stammen er starten af et CNN som oftest har en anden struktur end resten.

Grunden til det er at man fandt ud af at størstedelen af beregning sker i de første lag. Derfor prøver man at holde stammen så simpelt og hurtigt som muligt

1. **How does a bottleneck block differ from a plain ResNet block?**

Bottleneck layers bruger 3 forskellige convolutions 2 1x1 i begyndelsen og slutningen, og en 3x3, istedet for 2 3x3.

1. **Why is a bottleneck block faster?**

Bottleneck blocks reducerer antallet af parametre og matrix multiplications.

Hvis de 50 minutter ikke er gået så besvar følgende:

1. What is "adaptive pooling"?
2. What is "average pooling"?
3. Why do we need Flatten after an adaptive average pooling layer?
4. Why do we use plain convolutions in the CNN stem, instead of ResNet blocks?
5. How do fully convolutional nets (and nets with adaptive pooling in general) allow for progressive resizing?

Uge 17

Opgave 1

I skal kode en callbackfunktion, og anvende den. Jeg har fundet følgende ønske på nettet, som jeg synes kunne være et godt eksempel:

*use a condition to change a parameter (for instance change the learning rate if the validation loss hasn't improved)*

Uge 18

Opgave 1: Done

1. **Write the Python code to implement ReLU.**

if a > 0: return a else: return 0

1. **What is the "hidden size" of a layer?**

Repræsenterer antallet af neuroner i hvert lag i de skjulte lag.

1. **What does the t method do in PyTorch?**

Transpose

1. **Why is matrix multiplication written in plain Python very slow?**

For loops for daaays

Pytorch bruger C++ frem for Python, da Python er et langsomt sprog.

1. **In matmul, why is ac==br?**

Der er to matricer. Førset matrices kolonner (output?) skal være samme størrelse som anden matrices rækker (Input?)

og hvorfor? måske fordi det ellers ikke virker?

1. **In Jupyter Notebook, how do you measure the time taken for a single cell to execute?**

%time

1. **What is "elementwise arithmetic"?**

basis operationer bliver lavet for hvert enkelt element hvis begge tensor har samme shape.

Virker kun med samem shape, eller når de er broadcastable

1. **Write the PyTorch code to test whether every element of a is greater than the corresponding element of b.**

(a > b).all()

1. **What is a rank-0 tensor? How do you convert it to a plain Python data type?**

sum() og mean() er en reducerings operationer, som retunerer tensors med kun et element, som kunne være en boolean eller et nummer. Hvis man vil konventere det til plain python, skal man kalde på funktionen .item

(a + b).mean().item()

1. **What does this return, and why? tensor([1,2]) + tensor([1])**

Det retunerer: tensor([2,3]) da de er broadcastable og dermed bliver denne tensor på højre side lavet større.

1. **What does this return, and why? tensor([1,2]) + tensor([1,2,3])**

RuntimeError, da de ikke har samme størrelse og ikke er broadcastable

1. **How does elementwise arithmetic help us speed up matmul?**

Vi kan slette et loop ved at gange noget med noget andet

I Stedet for at lave den sidste operation manuelt med loop nummer 3, hvor vi tager hvert element af de to tensor og ganger det ene med det andet, gør vi brug af elementwise arithmetic, som laver præcis det internt. Derfor kan der slettes et loop

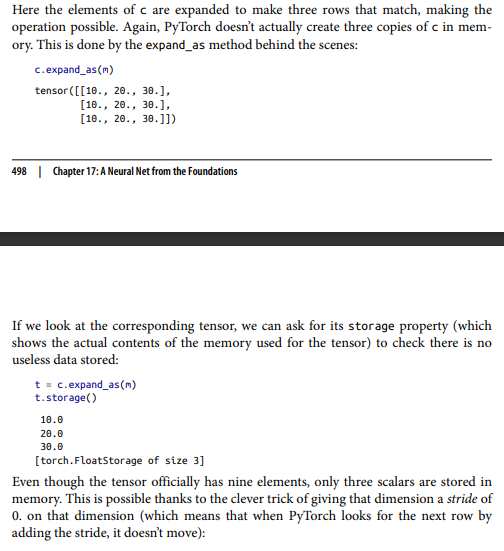
Black magic og matematik!

1. **What are the broadcasting rules?**

PyTorch sammenligner formen af elementer. Starter med at bruge *trailing dimensions* og arbejder baglæns. 2 dimensioner er kompatible når:

* 1. De er lige (ens)
  2. En af dem er 1, hvor i det tilfælde den dimension er broadcast til at være det samme som den anden.

1. **What is expand\_as? Show an example of how it can be used to match the results of broadcasting.**



1. **How does unsqueeze help us to solve certain broadcasting problems?**

Det giver muligheden for at broadcaste i en anden dimension, altså kolonner i stedet for rækker.

1. **How do we show the actual contents of the memory used for a tensor?**

tensor.storage()

1. **When adding a vector of size 3 to a matrix of size 3×3, are the elements of the vector added to each row or each column of the matrix? (Be sure to check your answer by running this code in a notebook.)**

Ret sikker på at det er på rows

1. **Do broadcasting and expand\_as result in increased memory use? Why or why not?**

Nej fordi at selve broadcastet ikke gemmes i en tensor i hukommelsen.

1. **What are the forward pass and backward pass of a neural network?**

Forward pass: Udregningen. Værdi af output fra input, udregner fra første neuron til sidste. Det er modellens gæt

Backward pass: Udregner forandring i vægtene using gradient descent. Backpropagation. Det er modellens forsøg i at blive klogere

1. **Why do we need to store some of the activations calculated for intermediate layers in the forward pass?**

Således at backward pass kan finde activation function

1. **What is the downside of having activations with a standard deviation too far away from 1?**

Activations ender enten med at blive nan eller 0

1. **How can weight initialization help avoid this problem?**

Ved at holde standard deviation af activations på 1, så man får brugbare tal.

1. **What is the formula to initialize weights such that we get a standard deviation of 1 for a plain linear layer, and for a linear layer followed by ReLU?**

Linear: 1/kvadrat rod(n\_in).

Med ReLU: 2/nin

1. **Why do we sometimes have to use the squeeze method in loss functions?**

For at fjerne dimensioner med en længde af 1. Altså er det en metode til at reshape en tensor.

1. **In what order do we need to call the \*\_grad functions in the backward pass? Why?**

mse, relu, lin

because chain rule

1. **What is \_\_call\_\_?**

Indbygget Python metode. Gør at man kan skrive klasser hvor instansen er en metode og kan blive kaldt som en metode

1. **If you are mathematically inclined, find out what the gradients of a linear layer are in mathematical notation. Map that to the implementation we saw in this chapter.**

Det er vi ikke.

1. **Implement everything in this chapter using NumPy instead of PyTorch…**

Boi, We have no time!