What else can we do?

- Data Augmentation
- Debugging Strategies
- Dropout
Data Augmentation

- Data augmentation is making the most of the training samples by introducing variations of these samples to accommodate for required invariances.
Data Augmentation

- Data augmentation is making the most of the training samples by introducing variations of these samples to accommodate for required invariances
- Why Data Augmentation?
Data Augmentation

Data augmentation is making the most of the training samples by introducing variations of these samples to accommodate for required invariances.

Why Data Augmentation?
  ▶ Because it’s all about the size of your data → More data for training
Data Augmentation

- Data augmentation is making the most of the training samples by introducing variations of these samples to accommodate for required invariances.

- Why Data Augmentation?
  - Because it’s all about the size of your data → More data for training
  - More importantly... to accommodate invariances
Invariances in data

- A *problem* is **invariant** to a *property* when the problem remains unchanged when transformations of a certain type are applied.
**Invariances in data**

- A *problem* is **invariant** to a *property* when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
<tr>
<td>Number plate recognition</td>
<td>translation, scaling</td>
</tr>
<tr>
<td>Action Recognition</td>
<td>translation, rotation, scaling, viewpoint, speed</td>
</tr>
</tbody>
</table>
Invariances in data

A problem is **invariant** to a *property* when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Invariances in data

- A problem is **invariant** to a property when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling</td>
</tr>
</tbody>
</table>
Invariances in data

- A *problem* is **invariant** to a *property* when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>
A problem is **invariant** to a property when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
<tr>
<td>Number plate recognition</td>
<td></td>
</tr>
</tbody>
</table>
A problem is invariant to a property when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
<tr>
<td>Number plate recognition</td>
<td>translation, scaling</td>
</tr>
</tbody>
</table>
A *problem* is **invariant** to a *property* when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
<tr>
<td>Number plate recognition</td>
<td>translation, scaling</td>
</tr>
<tr>
<td>Action Recognition</td>
<td></td>
</tr>
</tbody>
</table>
Invariances in data

A problem is **invariant** to a property when the problem remains unchanged when transformations of a certain type are applied.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
<tr>
<td>Number plate recognition</td>
<td>translation, scaling</td>
</tr>
<tr>
<td>Action Recognition</td>
<td>translation, rotation, scaling, viewpoint, speed</td>
</tr>
</tbody>
</table>

Dima Damen
Dima.Damen@bristol.ac.uk
COMSM0018: PRACTICAL - Lab4 - 2019/2020
Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

How to provide invariance?
Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

- How to provide invariance? → *artificially* augment for:
  - Translation:
  - Rotation:
  - Scaling:
  - Viewpoint:
Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

- How to provide invariance? → *artificially* augment for:
  - Translation: shifts – luckily CNNs do that for us
  - Rotation:
  - Scaling:
  - Viewpoint:
Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

How to provide invariance? → *artificially* augment for:

- Translation: shifts – luckily CNNs do that for us
- Rotation: rotations
- Scaling:
- Viewpoint:
### Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

- How to provide invariance? → *artificially* augment for:
  - Translation: shifts – luckily CNNs do that for us
  - Rotation: rotations
  - Scaling: cropplings
  - Viewpoint:
Invariances in data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Invariant to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Recognition</td>
<td>translation, rotation, scaling, viewpoint</td>
</tr>
</tbody>
</table>

- How to provide invariance? → *artificially* augment for:
  - Translation: shifts – luckily CNNs do that for us
  - Rotation: rotations
  - Scaling: croppings
  - Viewpoint: Minor - affine transformations, otherwise :-( collect more data!
Other invariances:
- invariance to random noise
- invariance to occlusion
- invariance to lighting conditions
- invariance to time of year!? — Generative!
Data Augmentation

- Data augmentation for invariances existed before deep learning

---


---

Dima Damen
Dima.Damen@bristol.ac.uk
COMS0018: PRACTICAL - Lab4 - 2019/2020
Why can’t current deep learning methods do that automatically for us?
But... Do I need more Data?

► There is a balance between the expense of collecting labelled data and refining the method.
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.
- How are you performing on your training data?
  - poorly
  - quite well
- I collected more data but things did not improve?
  - Think about the quality of your data.
  - Think about the quality of your labels.
  - fix and start over.
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.
- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have.
  - quite well.

Dima Damen
Dima.Damen@bristol.ac.uk
COMS0018: PRACTICAL - Lab4 - 2019/2020
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.
- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have.
  - quite well → How are you performing on your test data?
    - poorly
    - quite well
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method
- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have
  - quite well → How are you performing on your test data?
    - poorly → try augmentation, otherwise collect more data
    - quite well
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.

- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have.
  - quite well → How are you performing on your test data?
    - poorly → try augmentation, otherwise collect more data.
    - quite well → you’re done! Find a more interesting problem.
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.

- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have.
  - quite well → How are you performing on your test data?
    - poorly → try augmentation, otherwise collect more data.
    - quite well → you’re done! Find a more interesting problem.

- I collected more data but things did not improve?

Dima Damen
Dima.Damen@bristol.ac.uk
COMSM0018: PRACTICAL - Lab4 - 2019/2020
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method.

- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have.
  - quite well → How are you performing on your test data?
    - poorly → try augmentation, otherwise collect more data.
    - quite well → you’re done! Find a more interesting problem.

- I collected more data but things did not improve?
  - Think about the quality of your data.
But... Do I need more Data?

- There is a balance between the expense of collecting labelled data and refining the method
- How are you performing on your training data?
  - poorly → your algorithm needs work, you are not making the most of the data you have
  - quite well → How are you performing on your test data?
    - poorly → try augmentation, otherwise collect more data
    - quite well → you’re done! Find a more interesting problem
- I collected more data but things did not improve?
  - Think about the quality of your data
  - Think about the quality of your labels
But... Do I need more Data?

► There is a balance between the expense of collecting labelled data and refining the method

► How are you performing on your training data?
  ► poorly → your algorithm needs work, you are not making the most of the data you have
  ► quite well → How are you performing on your test data?
    ► poorly → try augmentation, otherwise collect more data
    ► quite well → you’re done! Find a more interesting problem

► I collected more data but things did not improve?
  ► Think about the quality of your data
  ► Think about the quality of your labels
  ► → fix and start over
Debugging Deep Learning Algorithms

- When a general machine learning code performs poorly, including deep learning code, it is very tricky to decide whether that is a bug in the code or a problem in the algorithm.
- Compiling correctly and getting numbers out is not an indication of correctness.
Debugging Deep Learning Algorithms

- When a general machine learning code performs poorly, including deep learning code, it is very tricky to decide whether that is a bug in the code or a problem in the algorithm.
- Compiling correctly and getting numbers out is not an indication of correctness.
- We do not know what the "correct" implementation will give in terms of accuracy, that is in fact what we wish to discover.
- Careful debugging is thus a must.
What is your baseline performance?

- Remember: you cannot perform worse than the baseline!
What is your baseline performance?

- Remember: you cannot perform worse than the baseline!
- What can an algorithm that makes decisions by “chance” do?

For a binary classifier, your baseline is 50%.

For a classifier into $N$ balanced classes, your baseline is $\frac{1}{N}\%$.

For a classifier with unbalanced classes...
What is your baseline performance?

- Remember: you cannot perform worse than the baseline!
- What can an algorithm that makes decisions by “chance” do?
- For a binary classifier, your baseline is 50%
What is your baseline performance?

- Remember: you cannot perform worse than the baseline!
- What can an algorithm that makes decisions by “chance” do?
- For a binary classifier, your baseline is 50%
- For a classifier into $N$ balanced classes, your baseline is $\frac{1}{N} \%$
What is your baseline performance?

- Remember: you cannot perform worse than the baseline!
- What can an algorithm that makes decisions by “chance” do?
- For a binary classifier, your baseline is 50%
- For a classifier into $N$ balanced classes, your baseline is $\frac{1}{N}\%$
- For a classifier with unbalanced classes??
Debugging Strategies

- *Mickey Mouse Examples* - test your solution on small tests that you know the outcome for
Debugging Strategies

- *Mickey Mouse Examples* - test your solution on small tests that you know the outcome for
- Evaluate the performance of your building blocks in isolation
Debugging Strategies

- *Mickey Mouse Examples* - test your solution on small tests that you know the outcome for
- Evaluate the performance of your building blocks in isolation
- Monitor the model in action
Debugging Strategies

- *Mickey Mouse Examples* - test your solution on small tests that you know the outcome for
- Evaluate the performance of your building blocks in isolation
- Monitor the model in action
- Look at failure cases (qualitative assessment)
Debugging Strategies

- *Mickey Mouse Examples* - test your solution on small tests that you know the outcome for
- Evaluate the performance of your building blocks in isolation
- Monitor the model in action
- Look at failure cases (qualitative assessment)
- Checkpoints and model saving
Debugging Strategies

“Directly observing the machine learning model performing its task will help to determine whether the quantitative performance numbers it achieves seem reasonable”

Goodfellow et al, p432
“Evaluation bugs can be some of the most devastating bugs because they can mislead you into believing your system is performing well when it is not”

Goodfellow et al, p432
Dropout

► What is regularisation?
► Remind yourself about dropout, as a regularisation/ensemble approach, from the lectures.
By the end of these lab sessions, you should be able to...

- Define a Fully-Connected Deep Neural Network (DNN) architecture
- Define a shallow Convolutional Neural Network (CNN) architecture
- Train and validate a CNN, and monitor its progress and results using Tensorboard
- Understand and estimate the effect of changing hyper-parameters on your results
- Implement and evaluate a variety of data augmentation techniques
- Implement dropout as one of the most common regularisation approaches
By the end of these lab sessions, you should be able to...

- Define a Fully-Connected Deep Neural Network (DNN) architecture
By the end of these lab sessions, you should be able to...

- Define a Fully-Connected Deep Neural Network (DNN) architecture
- Define a shallow Convolutional Neural Network (CNN) architecture
- Train and validate a CNN, and monitor its progress and results using Tensorboard
- Understand and estimate the effect of changing hyper-parameters on your results
- Implement and evaluate a variety of data augmentation techniques
- Implement dropout as one of the most common regularisation approaches
By the end of these lab sessions, you should be able to...

▶ Define a Fully-Connected Deep Neural Network (DNN) architecture
▶ Define a shallow Convolutional Neural Network (CNN) architecture
▶ Train and validate a CNN, and monitor its progress and results using Tensorboard
By the end of these lab sessions, you should be able to...

- Define a Fully-Connected Deep Neural Network (DNN) architecture
- Define a shallow Convolutional Neural Network (CNN) architecture
- Train and validate a CNN, and monitor its progress and results using Tensorboard
- Understand and estimate the effect of changing hyper-parameters on your results
By the end of these lab sessions, you should be able to...

▶ Define a Fully-Connected Deep Neural Network (DNN) architecture
▶ Define a shallow Convolutional Neural Network (CNN) architecture
▶ Train and validate a CNN, and monitor its progress and results using Tensorboard
▶ Understand and estimate the effect of changing hyper-parameters on your results
▶ Implement and evaluate a variety of data augmentation techniques
By the end of these lab sessions, you should be able to...

- Define a Fully-Connected Deep Neural Network (DNN) architecture
- Define a shallow Convolutional Neural Network (CNN) architecture
- Train and validate a CNN, and monitor its progress and results using Tensorboard
- Understand and estimate the effect of changing hyper-parameters on your results
- Implement and evaluate a variety of data augmentation techniques
- Implement dropout as one of the most common regularisation approaches
By the end of these lab sessions, you should be able to...

▶ By the end of this lab, you can upload all your zip files to Blackboard:
  ▶ Lab1_username.zip
  ▶ Lab2_username.zip
  ▶ Lab3_username.zip
  ▶ Lab4_username.zip

Deadline is 12th of November - but you can do it asap.
These will be marked for completion and originality - no judgement on any choices you made.

Labs should be individual work. You can use your code (any or all the group members) to start your project.
By the end of these lab sessions, you should be able to...

- By the end of this lab, you can upload all your zip files to Blackboard:
  - Lab1_username.zip
  - Lab2_username.zip
  - Lab3_username.zip
  - Lab4_username.zip
- Deadline is 12th of November - **but you can do it asap**
By the end of these lab sessions, you should be able to...

- By the end of this lab, you can upload all your zip files to Blackboard:
  - Lab1_username.zip
  - Lab2_username.zip
  - Lab3_username.zip
  - Lab4_username.zip
- Deadline is 12th of November - **but you can do it asap**
- These will be marked for completion and originality - no judgement on any choices you made.
By the end of these lab sessions, you should be able to...

- By the end of this lab, you can upload all your zip files to Blackboard:
  - Lab1_username.zip
  - Lab2_username.zip
  - Lab3_username.zip
  - Lab4_username.zip

- Deadline is 12th of November - **but you can do it asap**
- These will be marked for completion and originality - no judgement on any choices you made.
- Labs should be individual work. You can use your code (any or all the group members) to start your project.
And now....

READY....

STEADY....

GO....