Validation set vs training set LDA training - Approach two
CIFAR10

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
library("ggpubr")
```

```
## Warning: package 'ggpubr' was built under R version 4.0.5
```

This experiment focuses on the question, whether training LDA on the same set of data as the neural networks were trained on has any adverse effects to the performance of the ensemble as opposed to training on a different set, not presented to the networks during the training. Experiment was performed with two slightly different approaches.

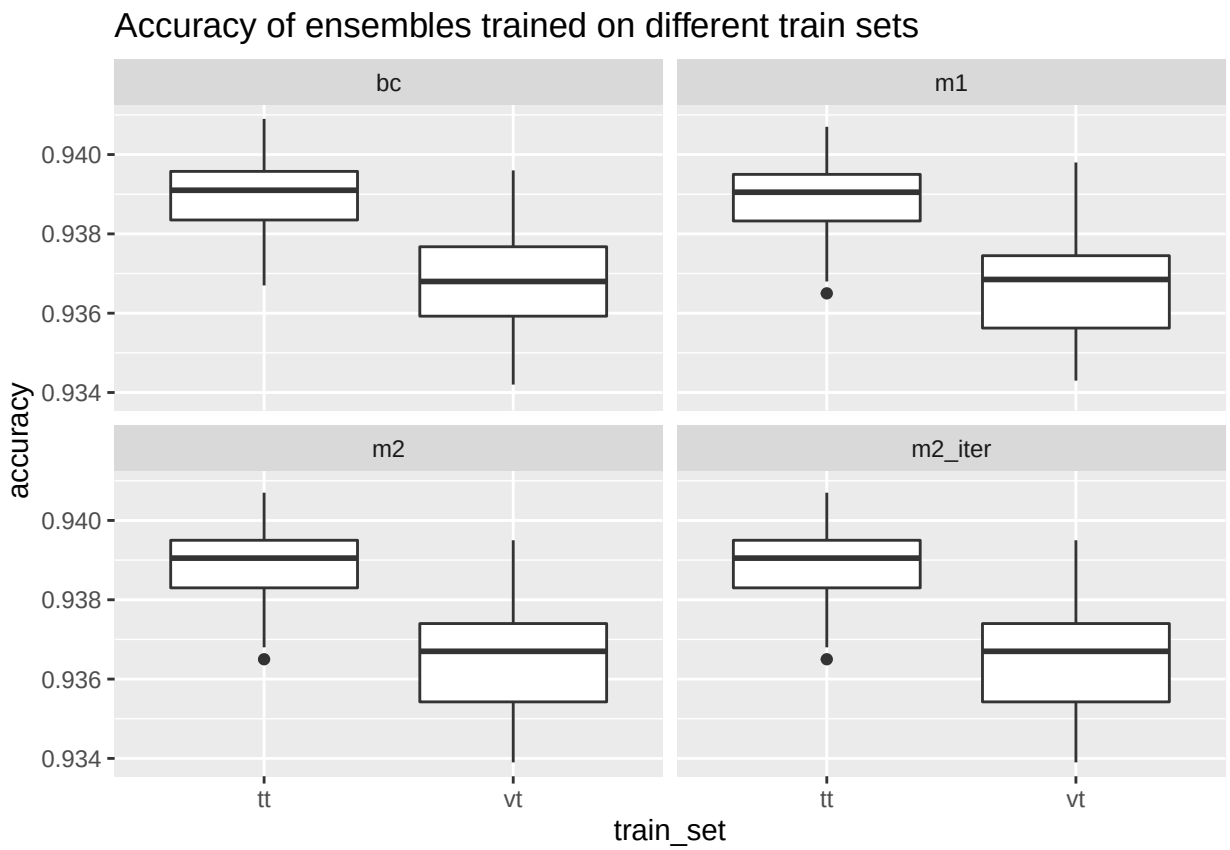
Approach two Experiment code is in the file half_train_ensembling_experiment.py. Experiment on both CIFAR10 and CIFAR100 datasets. This experiment differs from the previous one in the neural networks training part. In this case, the networks were trained on half of the original training set. The remainder of the training set was extracted as a validation set. This enabled us to train several ensembles on both the training set and the validation set in each replication. Experiment on CIFAR10 was performed in 1 replication and experimennt on CIFAR100 in 10 replications. This is due to 10 times more classes in CIFAR100 and thus a need for 10 times larger LDA training set in order to maintain constant 50 samples for class in LDA models training.

Approach one is described in the file visualization_base_experiment.

CIFAR10

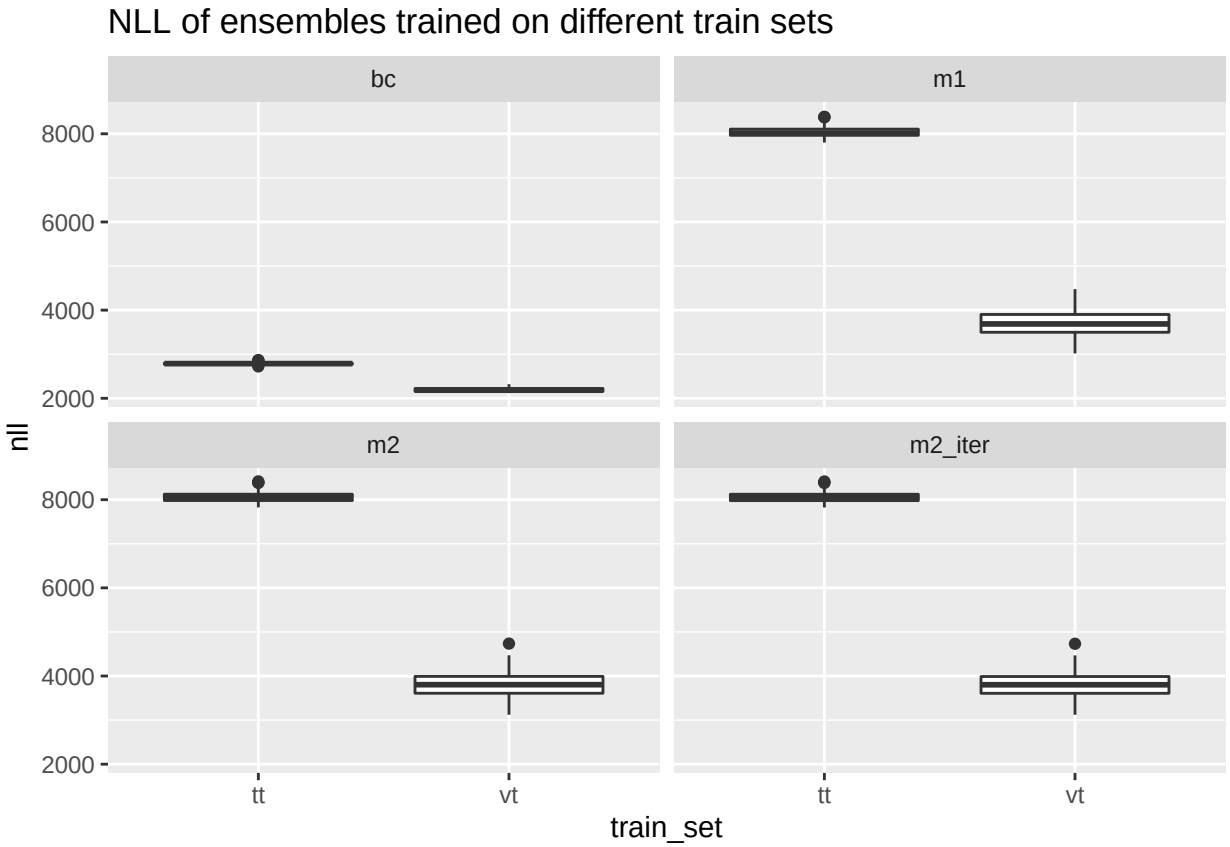
```
net_results <- read.csv("../data/data_train_val_half_c10/net_accuracies.csv")
ens_results <- read.csv("../data/data_train_val_half_c10/ensemble_accuracies.csv")
```

```
box_acc <- ggplot() + geom_boxplot(data=ens_results, mapping = aes(x=train_set, y=accuracy)) + facet_wra
  ggtitle("Accuracy of ensembles trained on different train sets")
box_acc
```



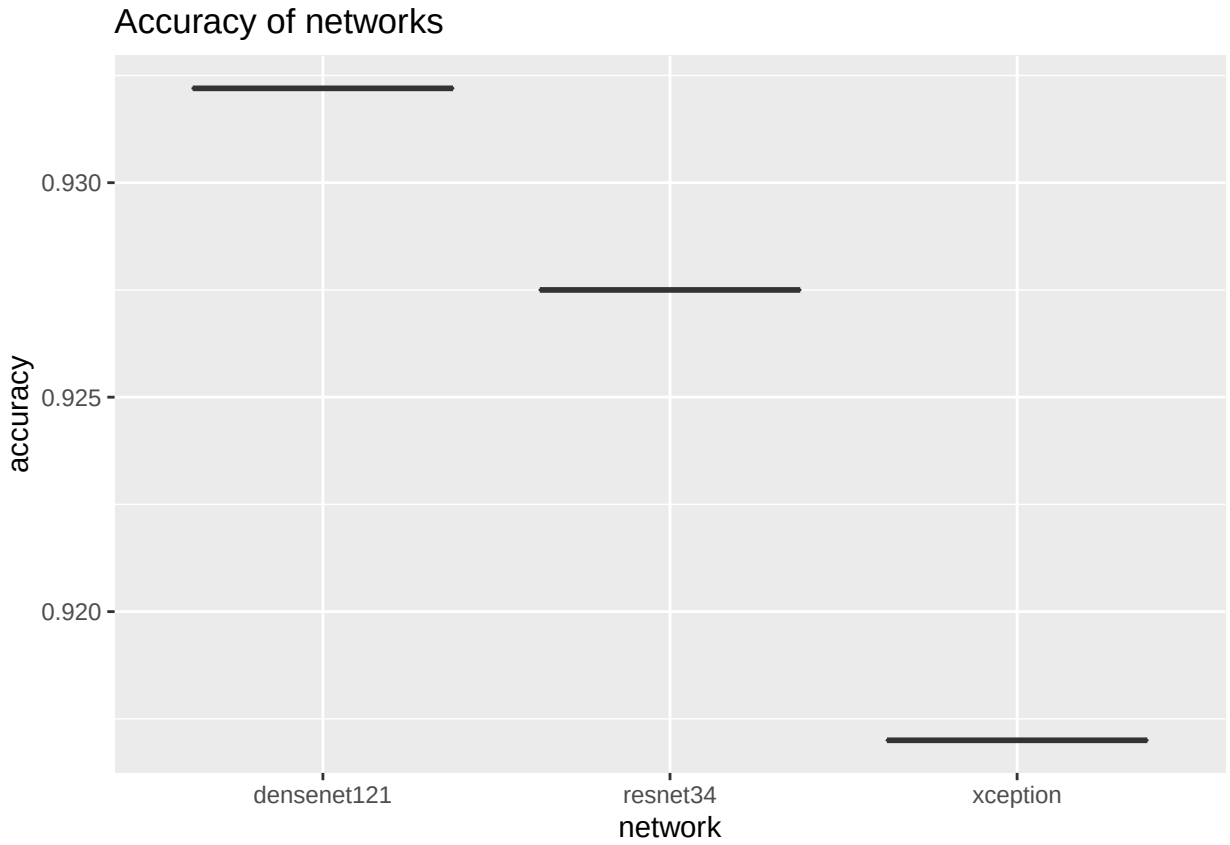
Training on the same training data as the neural networks were trained on seems to provide better accuracy compared to training on separate validation set.

```
box_nll <- ggplot() + geom_boxplot(data=ens_results, mapping = aes(x=train_set, y=nll)) + facet_wrap(~m
  ggtitle("NLL of ensembles trained on different train sets")
box_nll
```

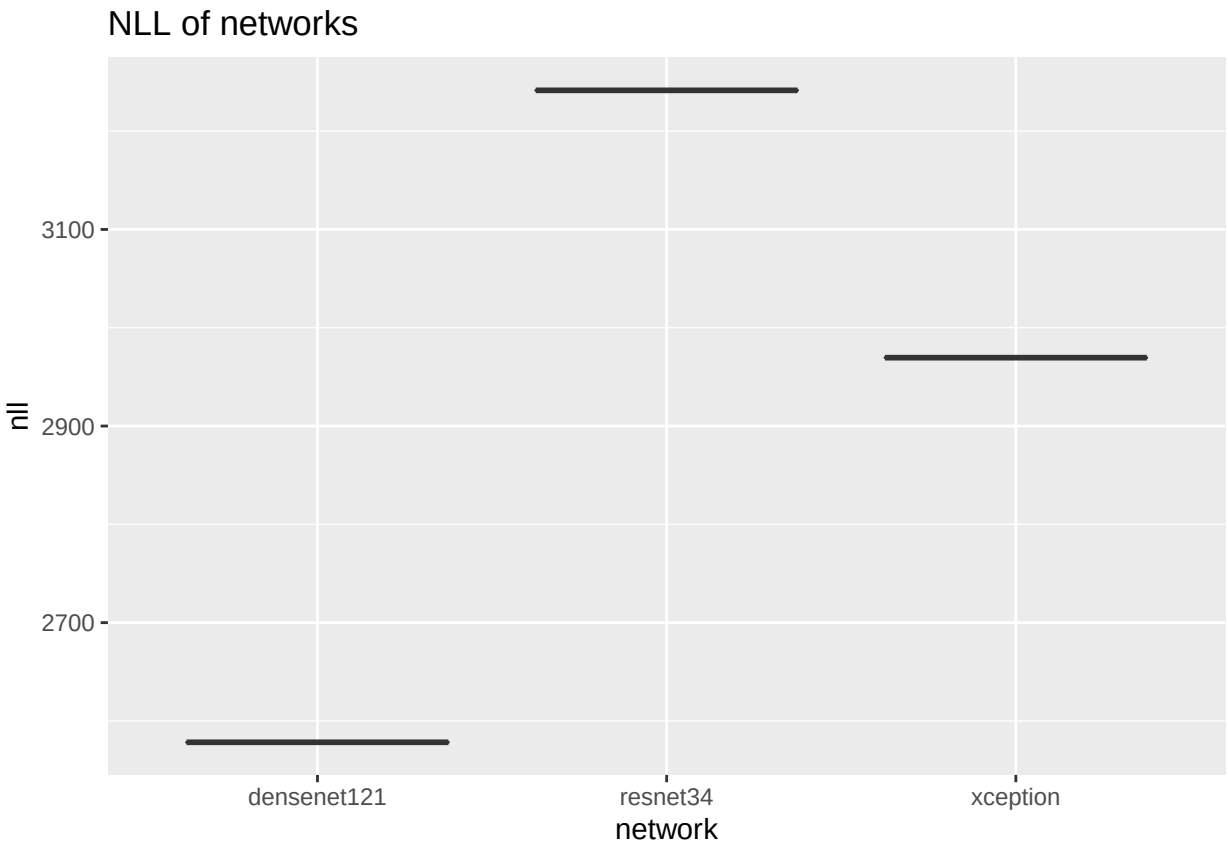


For some reason NLL has opposite trend to accuracy - train set with better accuracy has worse (larger) NLL. This needs some further attention.

```
box_net_acc <- ggplot() + geom_boxplot(data=net_results, mapping=aes(x=network, y=accuracy)) +
  ggtitle("Accuracy of networks")
box_net_acc
```




```
box_net_nll <- ggplot() + geom_boxplot(data=net_results, mapping=aes(x=network, y=nll)) +
  ggtitle("NLL of networks")
box_net_nll
```




```
ens_results_wide <- pivot_wider(ens_results, names_from = c(train_set, method), values_from = c(accuracy
ens_results_wide
```

```
## # A tibble: 50 x 18
##    repli  fold accuracy_tt_m1 accuracy_tt_m2 accuracy_tt_m2_iter accuracy_tt_bc
##    <int> <int>          <dbl>          <dbl>               <dbl>          <dbl>
##  1     0     0          0.940          0.939               0.939          0.940
##  2     0     1          0.940          0.940               0.940          0.94
##  3     0     2          0.939          0.939               0.939          0.939
##  4     0     3          0.938          0.938               0.938          0.938
##  5     0     4          0.939          0.939               0.939          0.939
##  6     0     5          0.940          0.940               0.940          0.940
##  7     0     6          0.939          0.939               0.939          0.939
##  8     0     7          0.940          0.940               0.940          0.940
##  9     0     8          0.939          0.939               0.939          0.940
## 10     0     9          0.937          0.937               0.937          0.938
## # ... with 40 more rows, and 12 more variables: accuracy_vt_m1 <dbl>,
## #   accuracy_vt_m2 <dbl>, accuracy_vt_m2_iter <dbl>, accuracy_vt_bc <dbl>,
## #   nll_tt_m1 <dbl>, nll_tt_m2 <dbl>, nll_tt_m2_iter <dbl>, nll_tt_bc <dbl>,
## #   nll_vt_m1 <dbl>, nll_vt_m2 <dbl>, nll_vt_m2_iter <dbl>, nll_vt_bc <dbl>
```

Testing statistical significance of difference for training on validation and train data. We have 50 samples from each distribution, so we will suppose the distributions are normal.

```r
t.test(ens_results_wide$accuracy_vt_m1, ens_results_wide$accuracy_tt_m1)
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_m1 and ens_results_wide$accuracy_tt_m1
## t = -9.5185, df = 86.022, p-value = 4.319e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002652213 -0.001735787
## sample estimates:
## mean of x mean of y
##  0.936698  0.938892
```

```r
t.test(ens_results_wide$accuracy_vt_m1, ens_results_wide$accuracy_tt_m1, alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_m1 and ens_results_wide$accuracy_tt_m1
## t = -9.5185, df = 86.022, p-value = 2.159e-15
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##          -Inf -0.001810737
## sample estimates:
## mean of x mean of y
##  0.936698  0.938892
```

```r
t.test(ens_results_wide$accuracy_vt_m2, ens_results_wide$accuracy_tt_m2)
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_m2 and ens_results_wide$accuracy_tt_m2
## t = -9.6417, df = 85.333, p-value = 2.617e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002718782 -0.001789218
## sample estimates:
## mean of x mean of y
##  0.936628  0.938882
```

```r
t.test(ens_results_wide$accuracy_vt_m2, ens_results_wide$accuracy_tt_m2, alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_m2 and ens_results_wide$accuracy_tt_m2
## t = -9.6417, df = 85.333, p-value = 1.308e-15
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
```

```
##          -Inf -0.001865254
## sample estimates:
## mean of x mean of y
##  0.936628  0.938882
```

```
t.test(ens_results_wide$accuracy_vt_m2_iter, ens_results_wide$accuracy_tt_m2_iter)
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_m2_iter and ens_results_wide$accuracy_tt_m2_iter
## t = -9.6417, df = 85.333, p-value = 2.617e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002718782 -0.001789218
## sample estimates:
## mean of x mean of y
##  0.936628  0.938882
```

```
t.test(ens_results_wide$accuracy_vt_bc, ens_results_wide$accuracy_tt_bc)
```

```
##
##  Welch Two Sample t-test
##
## data:  ens_results_wide$accuracy_vt_bc and ens_results_wide$accuracy_tt_bc
## t = -9.231, df = 82.425, p-value = 2.391e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002574402 -0.001661598
## sample estimates:
## mean of x mean of y
##  0.936856  0.938974
```

For all coupling methods, we found, that ensemble models trained on subset (of size 500) of the neural networks training set have statistically significantly better accuracy than ensemble models trained on separate validation set (of the same size).