

# Project 2

A comparison between Bayesian Neural Network and Classical Neural Network

Adrian Ispas

Grupa 507 - Master IA

<b>Informații generale</b>	<b>2</b>
<b>Sanity Check</b>	<b>3</b>
Simple Neural Network	3
Bayesian Neural Network	3
<b>Binary</b>	<b>6</b>
Neural Network	6
Bayesian Neural Network	6
<b>Multiclass</b>	<b>9</b>
Neural Network	9
Bayesian Neural Network	9
<b>Observații finale</b>	<b>12</b>

# Informații generale

Pe parcursul acestui document vor fi prezentate rezultatele cu observațiile asociate. Comentariile referitoare la implementare se găsesc în cod.

Implementare poate fi găsită atât pe github cu tagul asociat **v2.0.0** la următorul link: <https://github.com/adilspas/Bayesian-Neural-Network/releases/tag/v2.0.0> dar și atașată mail-ului trimis.

Tagul **v1.0.0** este asociat implementării în Edward.  
[<https://github.com/adilspas/Bayesian-Neural-Network/tree/v1.0.0>]

# Sanity Check

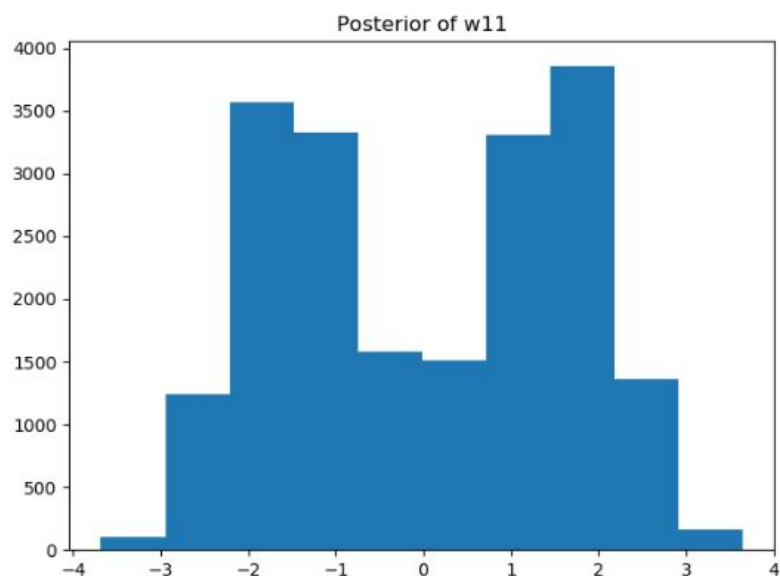
Pentru sanity check am folosit arhitectura simplă de rețea prezentată în curs atât pentru rețeaua neurală cât și pentru rețeaua bayesiană.

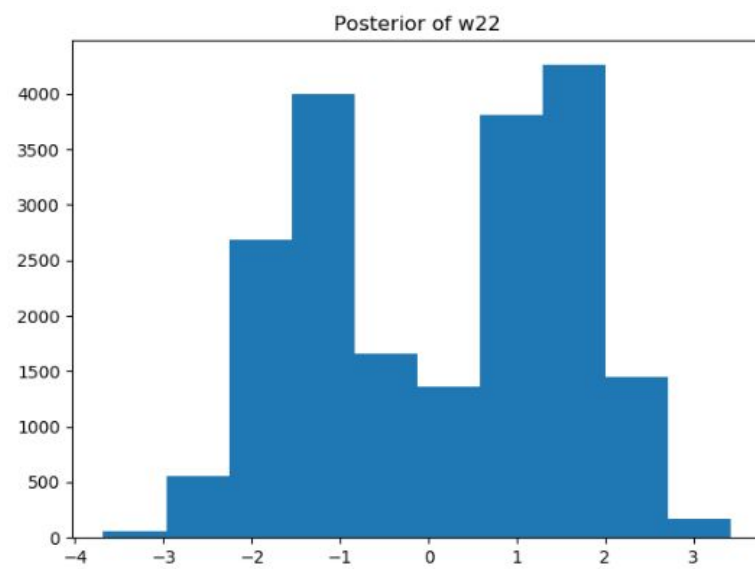
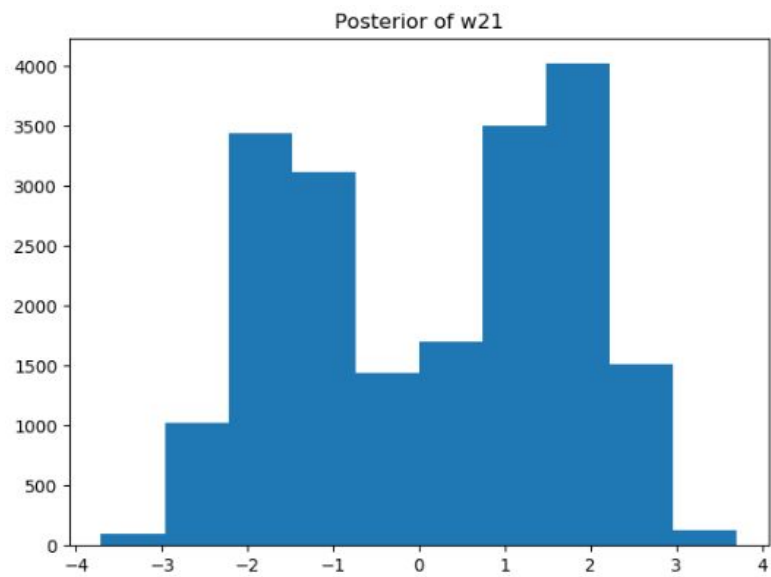
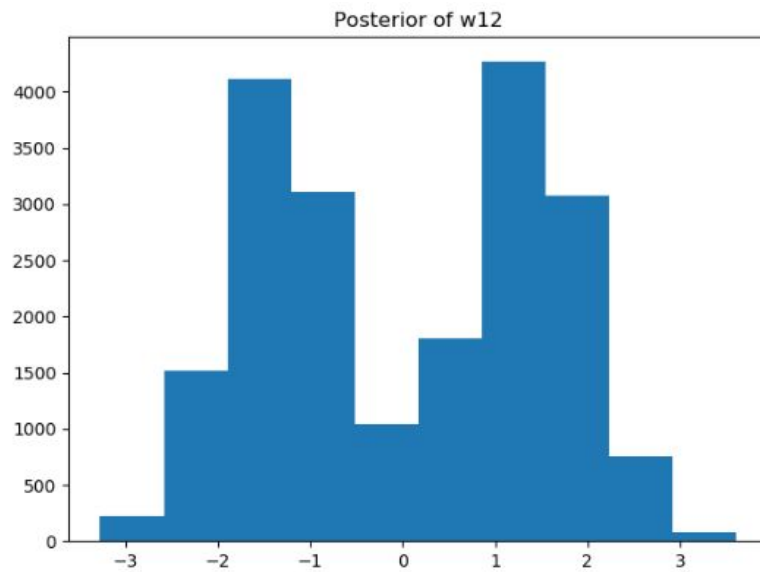
## Simple Neural Network

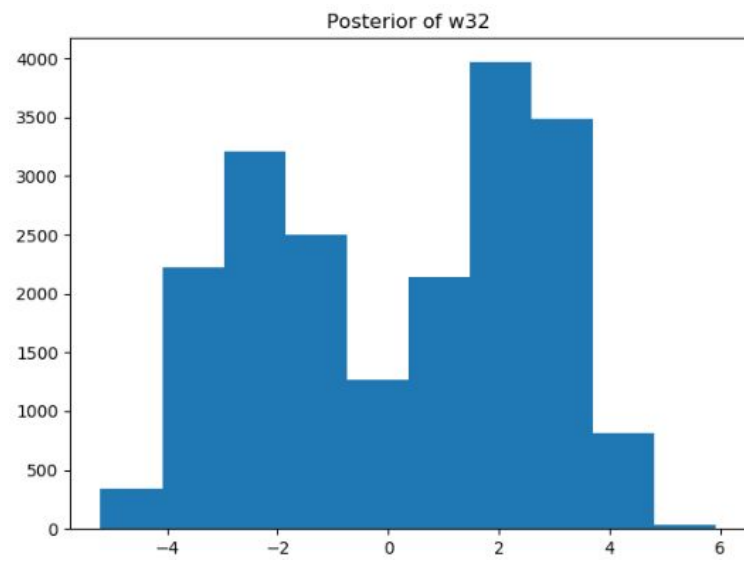
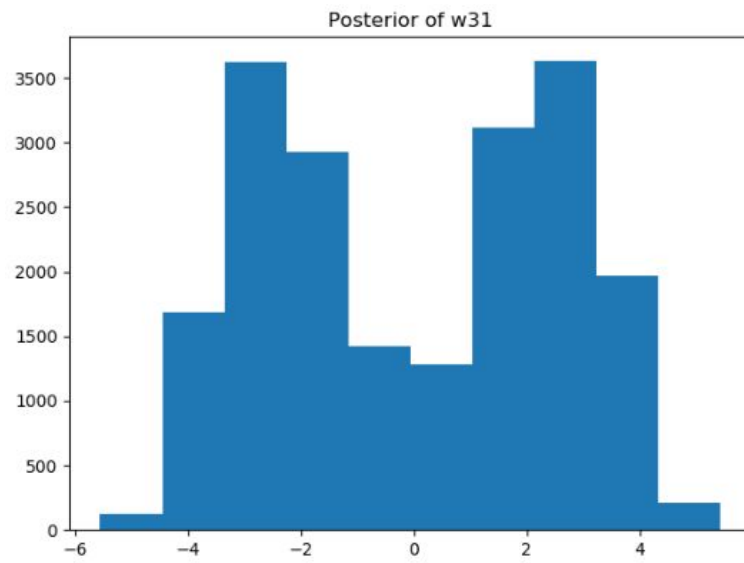
```
One sample of weights (w11, w12, w21, w22, w31, w32):  
9.686704932943504 3.2368500047231032 11.944259685745003 0.5193397295117469  
9.976973500819865 0.9210947925884191  
Accuracy on train data: 1.0  
Accuracy on test data: 0.78
```

## Bayesian Neural Network

```
One sample of weights (w11, w12, w21, w22, w31, w32):  
-1.4475066050801735 -1.6163287068514152 1.605436547421329 1.6443853501580756  
-3.3114385786076674 2.104616865902536  
Accuracy on train data: 0.98  
Accuracy on test data: 0.96
```







# Binary

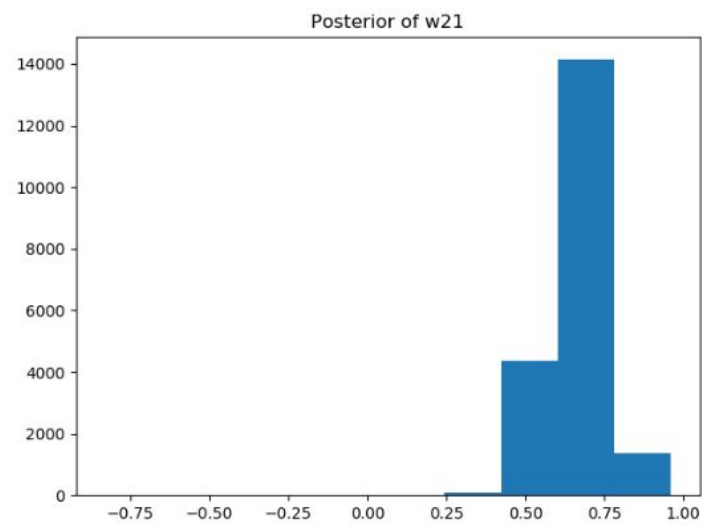
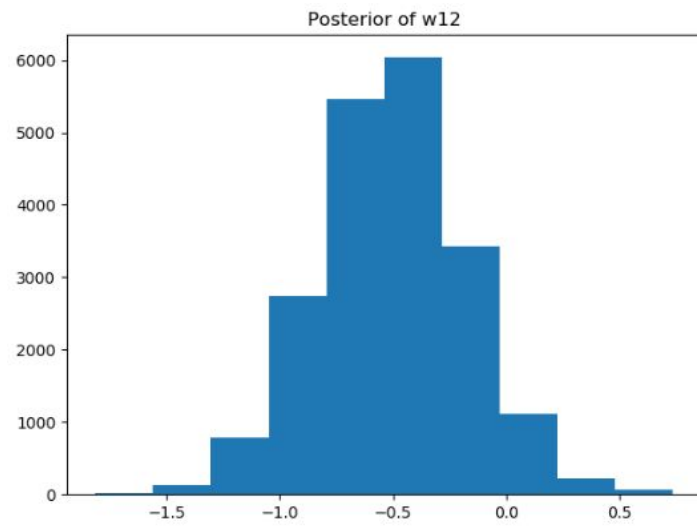
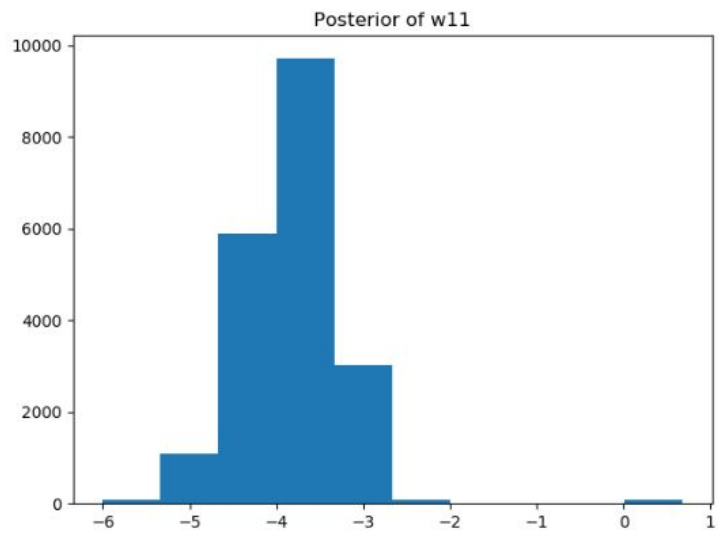
Pentru clasificarea binară am folosit în cazul rețelei neurale o rețea cu 3 layere, iar în cazul rețelei bayesiene rețeaua anterioară observând că are rezultate bune și în această situație.

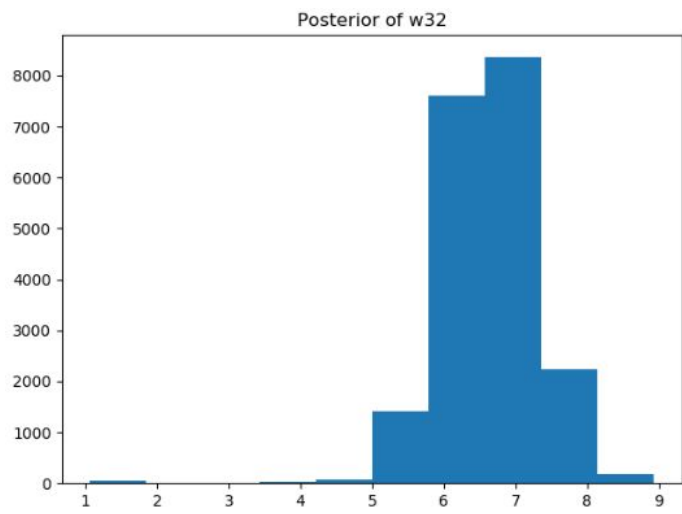
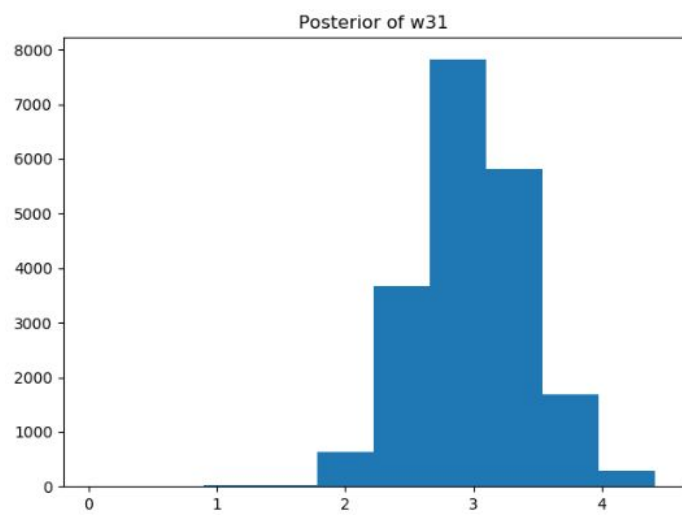
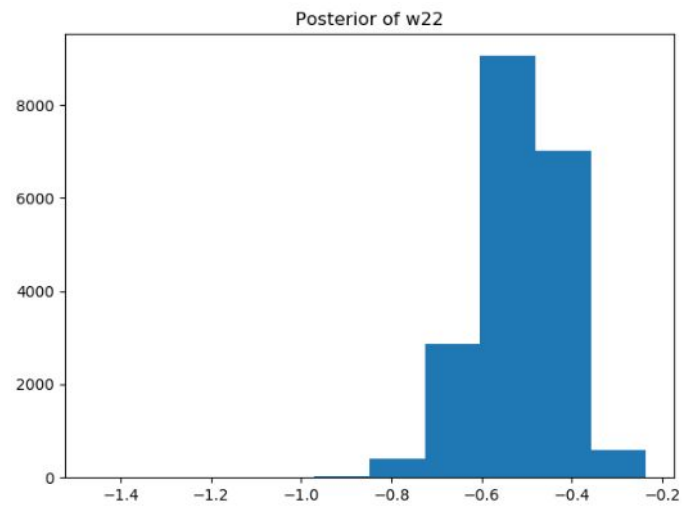
## Neural Network

```
Loss:0.5001262822840983
Loss:0.14999949934272314
Loss:0.5
Loss:0.12666666666666665
Loss:0.38247663773958523
Loss:0.12002139832639978
Loss:0.13333333333333333
Loss:0.12
Loss:0.12666778566632614
Loss:0.12664119511597852
Accuracy on train data: 0.86
Accuracy on test data: 0.5
```

## Bayesian Neural Network

```
[ -                    5%                ] 1025 of 20000 complete in 0.5 sec
[ ---                 10%                ] 2088 of 20000 complete in 1.0 sec
[ -----            15%                ] 3179 of 20000 complete in 1.5 sec
[ -----            21%                ] 4310 of 20000 complete in 2.0 sec
[ -----            26%                ] 5390 of 20000 complete in 2.5 sec
[ -----            32%                ] 6535 of 20000 complete in 3.0 sec
[ -----            38%                ] 7666 of 20000 complete in 3.5 sec
[ -----            43%                ] 8751 of 20000 complete in 4.0 sec
[ -----            49%                ] 9809 of 20000 complete in 4.5 sec
[ -----            54%                ] 10892 of 20000 complete in 5.0 sec
[ -----            59%--              ] 11964 of 20000 complete in 5.5 sec
[ -----            64%----            ] 12991 of 20000 complete in 6.0 sec
[ -----            70%-----          ] 14052 of 20000 complete in 6.5 sec
[ -----            75%-----          ] 15129 of 20000 complete in 7.0 sec
[ -----            81%-----          ] 16205 of 20000 complete in 7.5 sec
[ -----            86%-----          ] 17279 of 20000 complete in 8.0 sec
[ -----            91%-----          ] 18359 of 20000 complete in 8.5 sec
[ -----            97%-----          ] 19421 of 20000 complete in 9.0 sec
[ -----           100%-----          ] 20000 of 20000 complete in 9.3 sec
Accuracy on train data: 0.86
Accuracy on test data: 0.852
```







# Multiclass

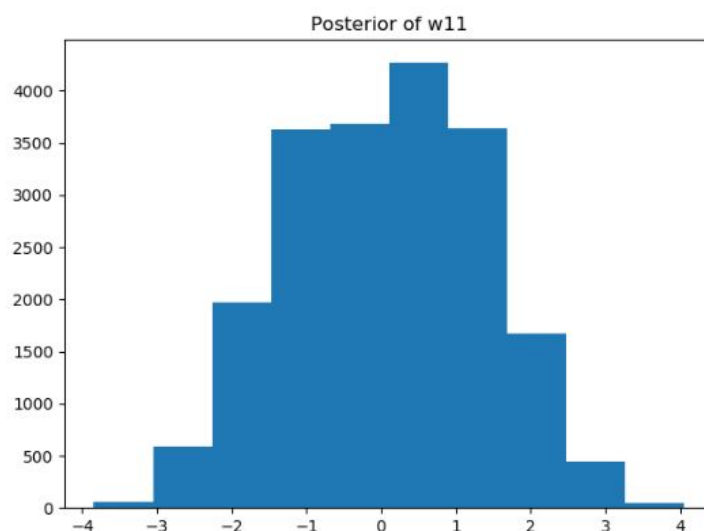
În cazul clasificării multclasă am folosit arhitecturi similare de rețele ca cele prezente la clasificarea binară.

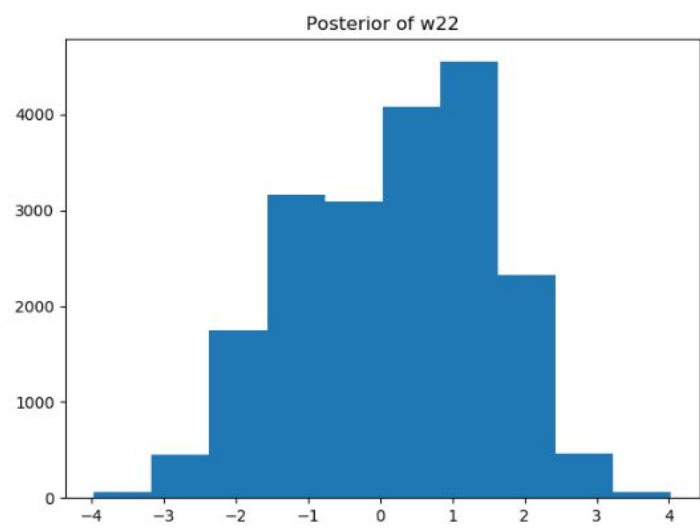
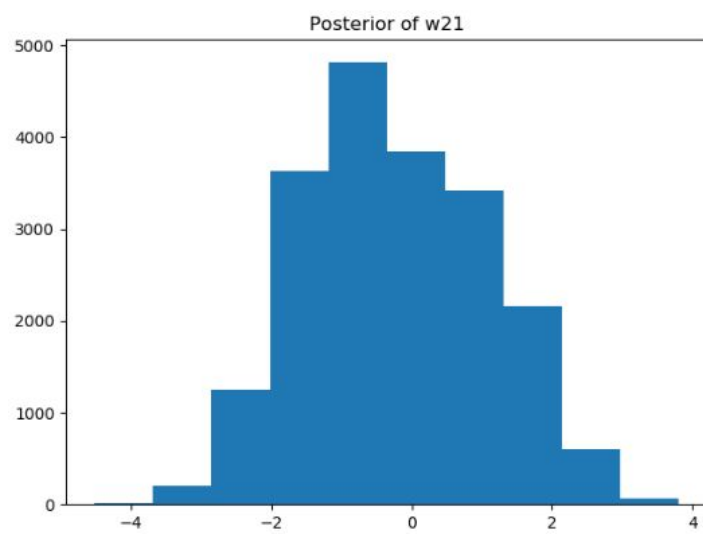
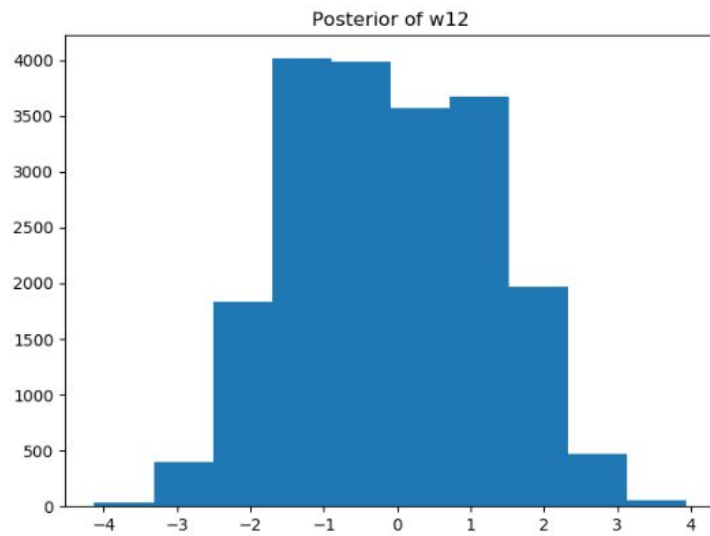
## Neural Network

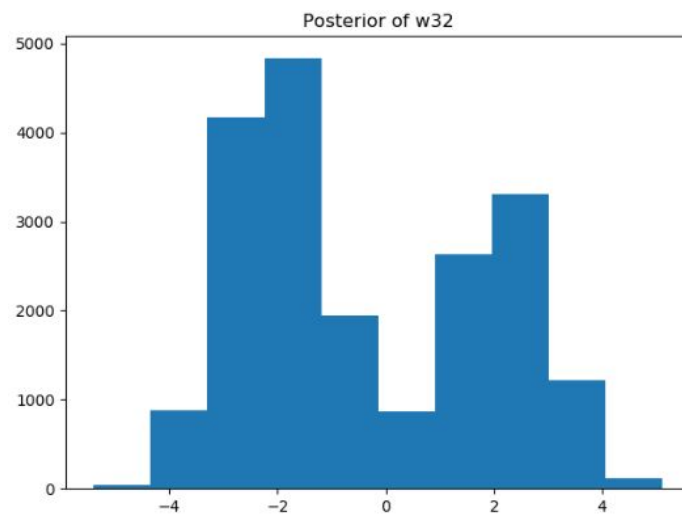
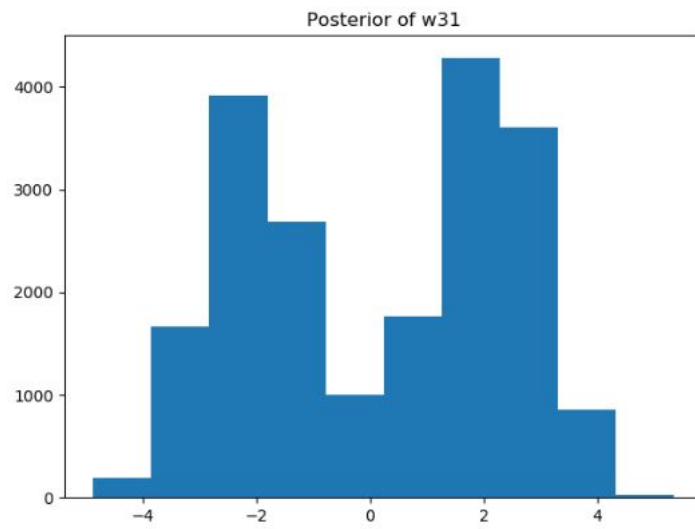
```
Loss:0.7012694472911734
Loss:0.3416009261061802
Loss:0.3799016721129189
Loss:0.42053117818372177
Loss:0.42053117818372177
Loss:0.42053117818372177
Loss:0.42053117818372177
Loss:0.42053117818372177
Loss:0.42053117818372177
Loss:0.42053117818372177
Accuracy on train data: 0.3466666666666667
Accuracy on test data: 0.32
```

## Bayesian Neural Network

```
[-----100%-----] 20000 of 20000 complete in 24.8 sec
Probabilities in categorical_like sum to [42.4219091]
Accuracy on train data = 0.36
Accuracy on test data = 0.3466666666666667
```



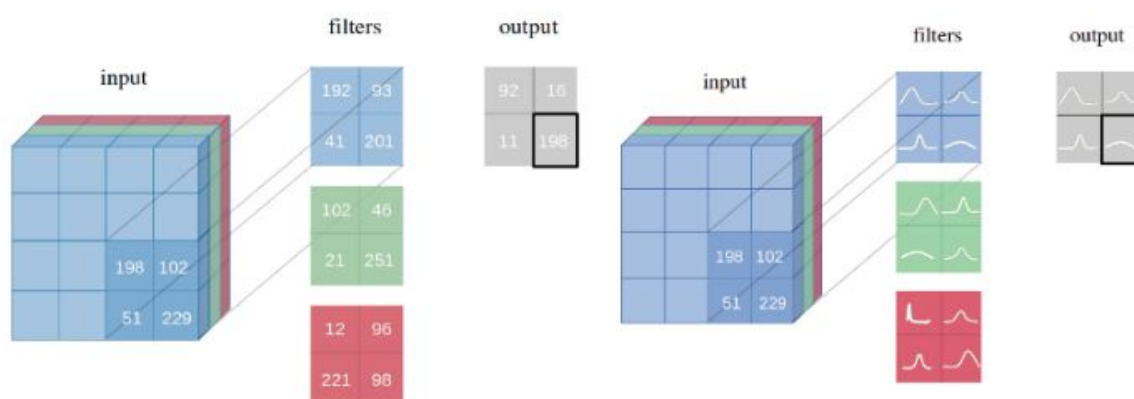




## Observații finale

În implementarea taskului am abordat o arhitectură simplă atât pentru rețelele neurale cât și pentru cele bayesiene, urmărind în principal prin trace pe  $\mathbf{w}$ -uri modelul pe care îl propun rețelele bayesiene și anume distribuții peste  $\mathbf{w}$ -uri în comparație cu rețelele „clasice” care oferă o singură estimare a  $\mathbf{w}$ -ului.

Idee care poate fi ușor intuită din imaginea de mai jos.



Sursa imagini: <https://medium.com/neuralspace/bayesian-convolutional-neural-networks-with-bayes-by-backprop-c84dcaaf086e>