

# Package 'DNNGP'

**Title** Deep neural network for genomic prediction

**Version** 1.0.0

**Maintainer** Huihui Li <lihuihui@caas.cn>, Kelin Wang <wangkelin\_2022@163.com>

**Description** The Python project 'DNNGP' can be used to implement genome-wide prediction (GP), which can predict the phenotypes of plants and animals based on multi-omics data. The code is written using Python 3.6 and TensorFlow 1.15.

**Depends** Python 3.6, TensorFlow 1.15

**License** GNU GPLv3

**Encoding** UTF-8

## Installation

Installation supports Python 3.6. Follow the instructions at <https://www.tensorflow.org/install/gpu> to set up GPU support for faster model training. Once GPU is set up, install with conda by executing these instructions from the root of the checked-out repository:

```
conda create -n dnngp python=3.6.5
```

```
conda activate dnngp
```

```
pip install -r requirements.txt
```

Users can also use DNNGP on CPU, and the installation method is the same as above.

## Examples

To run locally, there are two required input files. One file contains the phenotype of interest, the other file contains the SNP data, genomic expression data or other related omics data with digital coding.

An example command to train DNNGP to predict the phenotype from the SNP data, genomic expression data or other related omics data with digital coding is the following:

```
python dnngp_runner.py \  
  --batch_size [num] \  
  --epoch [num] \  
  --lr [num] \  
  --patience [num] \  
  --dropout1 [num] \  
  --dropout2 [num] \  
  --SNP [your omics-data file] \  
  --pheno [your phenotype data file]
```

Of particular note is the *run.py*. This script is used to get results in batches by adjusting different hyperparameters and inputs.

Each parameter and hyperparameter are explained as follows:

***batch\_size***: Number of samples selected in one training epoch

***epoch***: Use all the data in the training set to conduct a complete training for the model, and record it as an epoch

***lr***: learning rate

***patience***: Biggest number of epoch within the improvement of model performance observed, once there is no improvement in patience iterations, the learning rate will be reduced

***dropout1***: dropout rate of the first dropout layer

***dropout2***: dropout rate of the second dropout layer

***SNP***: your genomic input data

***pheno***: your phenotype input data

### **Data**

tomato332(/DNNGP-master/example-data/tomato332)

wheat599(/DNNGP-master/example-data/wheat599)

maize1404(/DNNGP-master/example-data/maize1404)

wheat2000(/DNNGP-master/example-data/wheat2000)