Hyperparameter tuning part III

Plan for the lecture: models

- Tree-based models
  - GBDT: `XGBoost, LightGBM, CatBoost`
  - RandomForest/ExtraTrees
- Neural nets
  - `Pytorch, Tensorflow, Keras...`
- Linear models
  - SVM, logistic regression
  - `Vowpal Wabbit, FTRL`
- Factorization Machines (out of scope)
  - `libFM, libFFM`

Plan for the lecture: models

- **What framework to use?**
  - Keras, Lasagne
  - TensorFlow
  - MxNet
  - PyTorch
  - sklearn's MLP
  - …

They implement the same functionality! (except sklearn)

- **I recommend:**
  - PyTorch
  - Keras

Neural Nets

Neural Nets

- Number of neurons per layer

Neural Nets

- Number of neurons per layer
- Number of layers

Neural Nets

- Number of neurons per layer
- Number of layers
- Optimizers
  - SGD + momentum
  - Adam/Adadelta/Adagrad/...
    - In practice lead to more overfitting

Neural Nets

- Number of neurons per layer
- Number of layers
- Optimizers
  - SGD + momentum
  - Adam/Adadelta/Adagrad/…
    - In practice lead to more overfitting
- Batch size

Neural Nets

- Number of neurons per layer
- Number of layers
- Optimizers
  - SGD + momentum
  - Adam/Adadelta/Adagrad/...
    - In practice lead to more overfitting
- Batch size

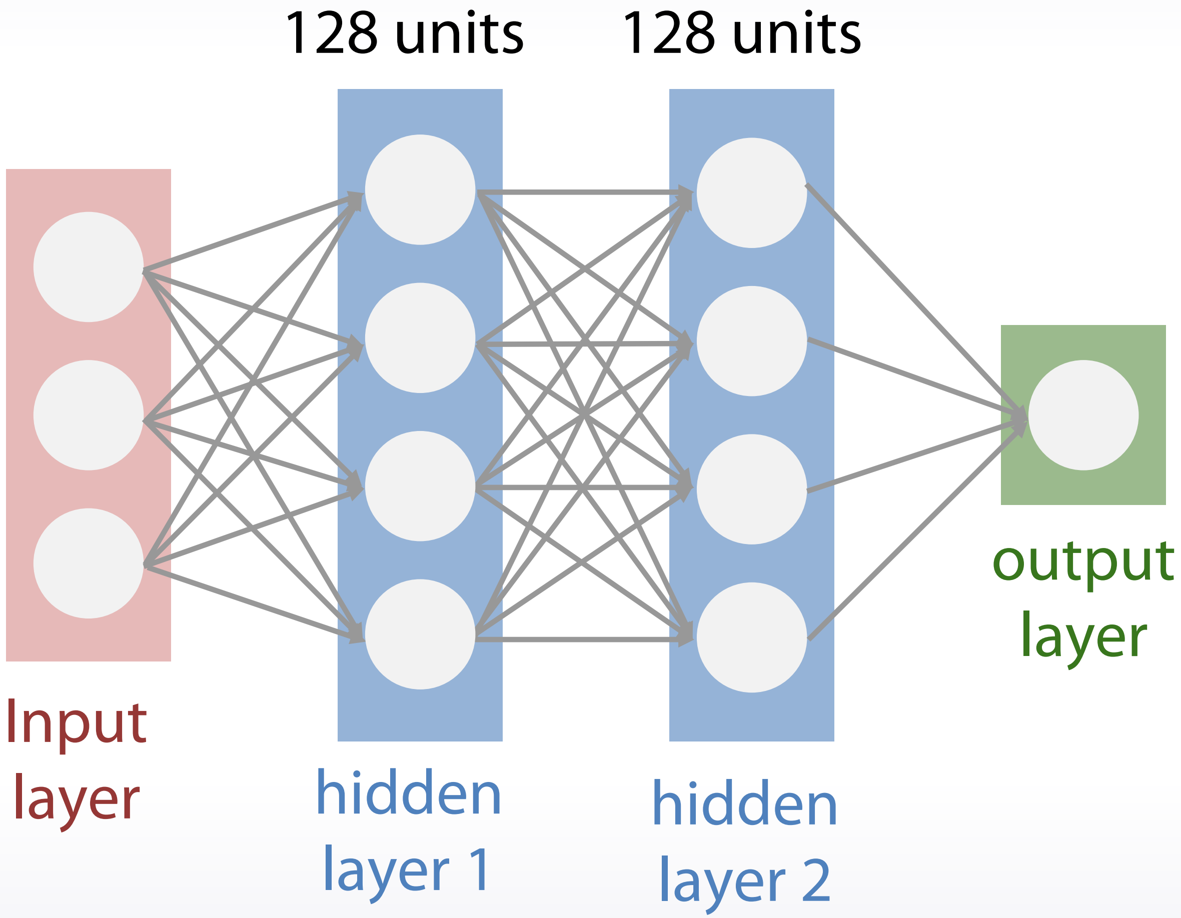
Neural Nets

- Number of neurons per layer
- Number of layers
- Optimizers
  - SGD + momentum
  - Adam/Adadelta/Adagrad/…
    - In practice lead to more overfitting
- Batch size
- Learning rate

Neural Nets

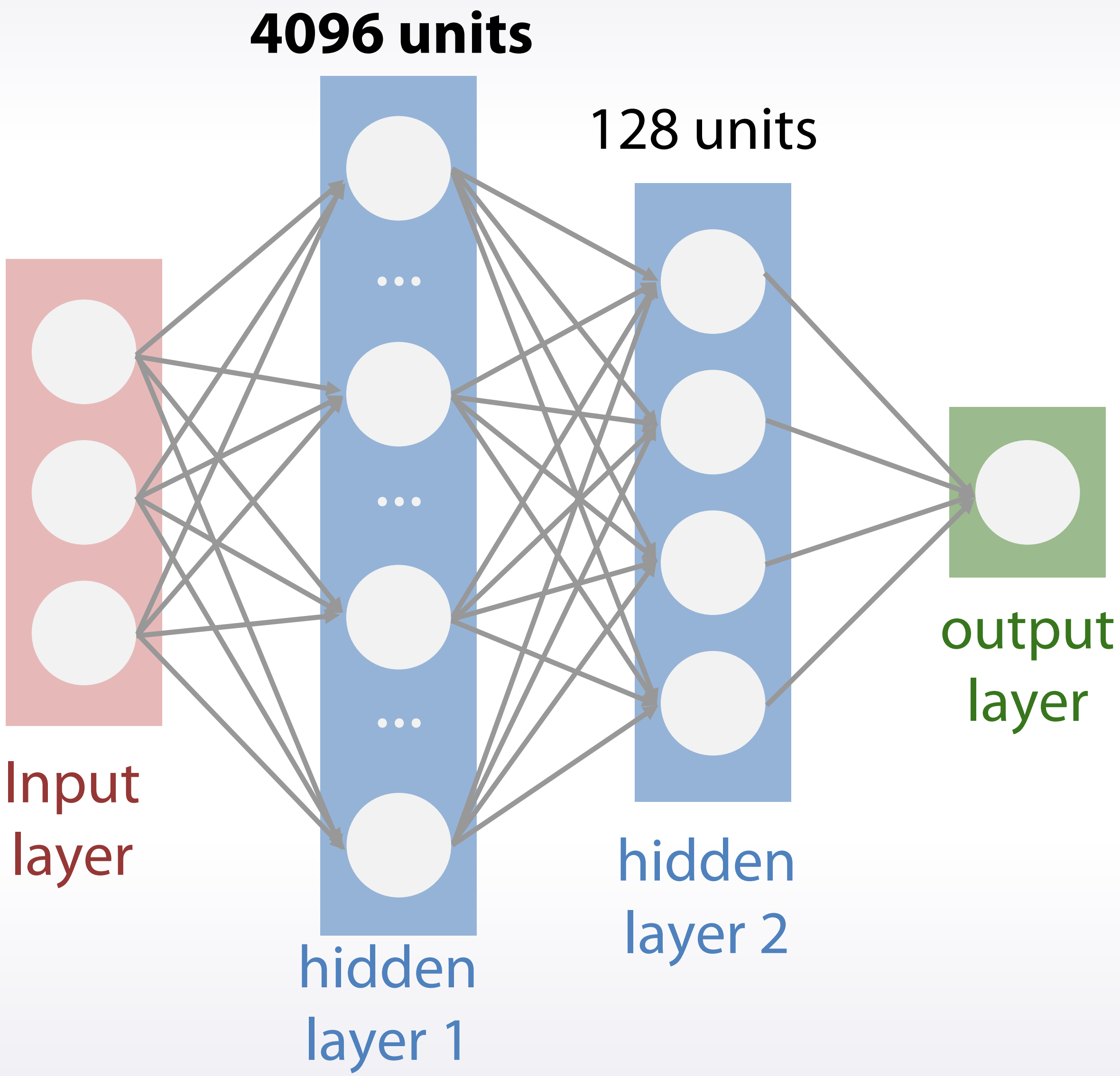
- Number of neurons per layer
- Number of layers
- Optimizers
  - SGD + momentum
  - Adam/Adadelta/Adagrad/...
    - In practice lead to more overfitting
- Batch size
- Learning rate
- Regularization
  - L2/L1 for weights
  - Dropout/Dropconnect
  - Static dropconnect

Static dropconnect

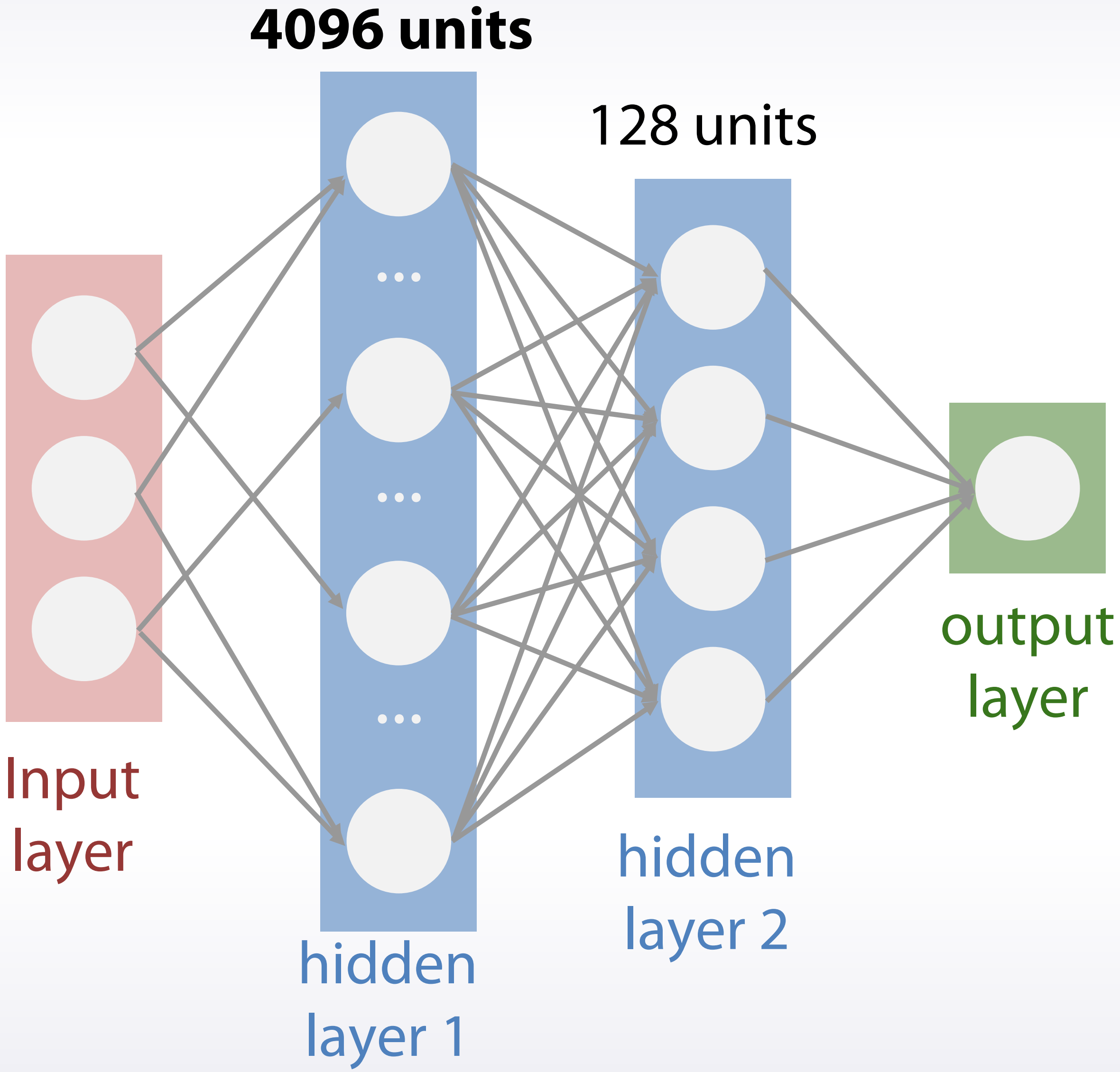


Based on image from *Stanford University CS231n*

Static dropconnect



Static dropconnect



Linear models

- **Scikit-learn**
  - `SVC/SVR`
    - Sklearn wraps `libLinear` and `libSVM`
    - Compile yourself for multicore support

Linear models

- **Scikit-learn**
  - `SVC/SVR`
    - Sklearn wraps `libLinear` and `libSVM`
    - Compile yourself for multicore support
  - `LogisticRegression/LinearRegression +` regularizers
  - `SGDClassifier/SGDRegressor`

Linear models

- **Scikit-learn**
  - `SVC/SVR`
    - Sklearn wraps `libLinear` and `libSVM`
    - Compile yourself for multicore support
  - `LogisticRegression/LinearRegression` + regularizers
  - `SGDClassifier/SGDRegressor`
- **`Vowpal Wabbit`**
  - `FTRL`

Linear models

- Regularization parameter (C, alpha, lambda, …)
  - Start with very small value and increase it.
  - SVC starts to work slower as C increases

Linear models

- Regularization parameter (C, alpha, lambda, …)
  - Start with very small value and increase it.
  - SVC starts to work slower as C increases

- Regularization type
  - *L1/L2/L1+L2* -- try each
  - *L1* can be used for feature selection

Tips

- **Don’t spend too much time tuning hyperparameters**
  - Only if you don’t have any more ideas or you have spare computational resources

Tips

- **Don’t spend too much time tuning hyperparameters**
  - Only if you don’t have any more ideas or you have spare computational resources
- **Be patient**
  - It can take thousands of rounds for GBDT or neural nets to fit

Tips

- **Don’t spend too much time tuning hyperparameters**
  - Only if you don’t have any more ideas or you have spare computational resources
- **Be patient**
  - It can take thousands of rounds for GBDT or neural nets to fit
- **Average everything**
  - Over random seed
  - Or over small deviations from optimal parameters
    - e.g. average *max_depth=4,5,6* for an optimal *5*

Conclusion

- Hyperparameter tuning in general
  - General pipeline
  - Manual and automatic tuning
  - What should we understand about hyperparameters?
- Models, libraries and hyperparameter optimization
  - Tree-based models
  - Neural networks
  - Linear models