CS5242 Deep Learning and Neural Network

Predicting Protein Ligand Interaction

Group 33

Wang Qijia

Gao Bin

Outline

- First thoughts and challenges
- Solution
- Training
- Testing
- Results
- Takeaways

First thoughts…

- CNN on protein and ligand
- Atomic CNN and radial pooling

Challenges

- Vast space and high precision of coordinates for atoms:
  - (-244.401, -229.648, -177.028) to (310.935, 432.956, 435.107)
  - Tensor size limit
  - RAM limit
- Not enough training examples
- Difficulties in implementation

Solution

Assumptions

- False negative
- Pairs in testing data

Techniques

- Distance filtering
- ROI pooling and voxelization
- CNN on substructure

Distance Filtering

- Ligand-protein distance

$$distance_{protein-ligand} = \max_i \left(\min_j distance_{i,j} \right)$$

Condition	Max Distance	Average Distance	Percentage
Binding	7	5	0.03%
Non-binding && distance > 10	610	53	82%
Non-binding && distance > 7	610	50.9	86.8%

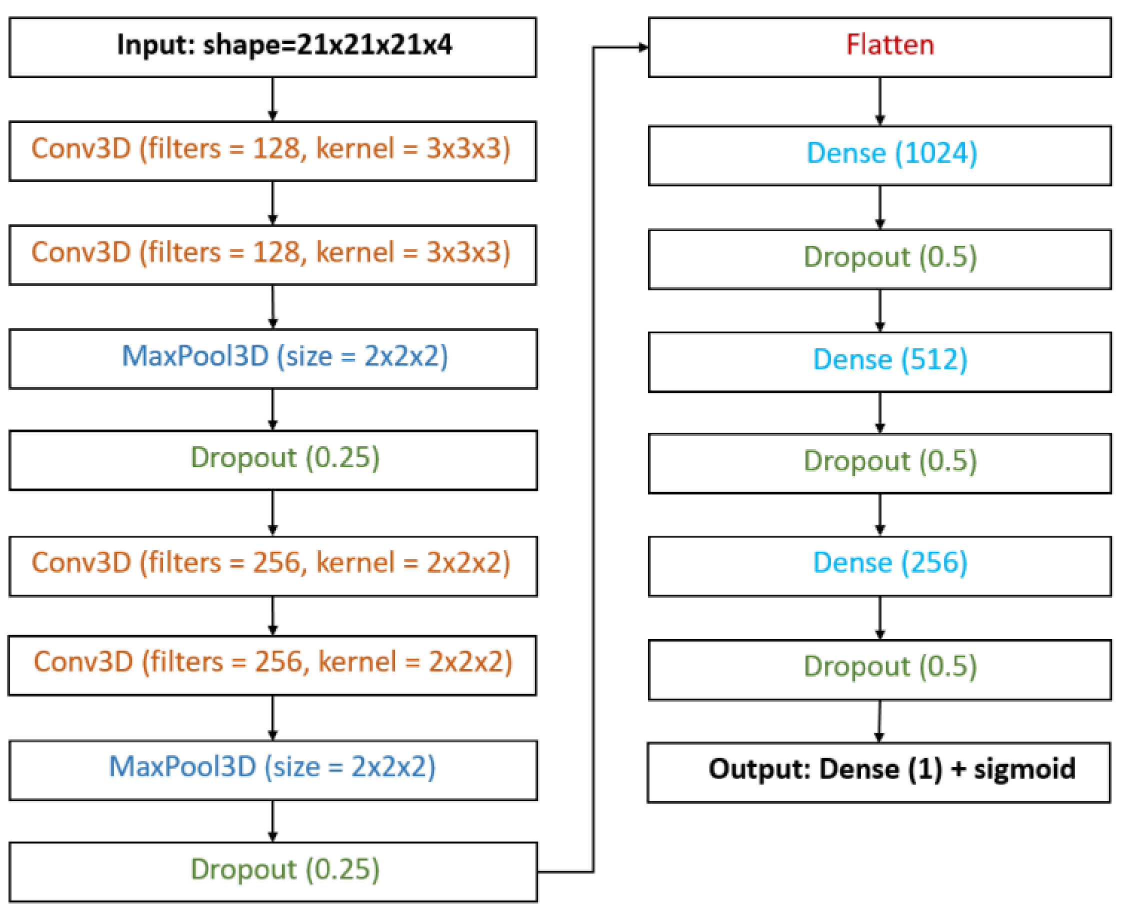
- Rule: Consider the pair as non-binding if distance is greater than 10

ROI pooling and voxelization

- Protein atoms outside ligand atom neighborhood should not have impact on the binding
- Each ROI is a small grid with a ligand atom in the center
- Voxelized grid
- 4 Channels (isProtein, proteinPolarity, isLigand, ligandPolarity)

- Other voxelization options (external libraries)

CNN model



Optimizer	Adam
Loss	Binary cross-entropy
Callback	Early-stopping
Batch size	100

But…

- The output from CNN model is per-ligand-atom score

- How to calculate the score for the protein-ligand pair?
  - Fully connected layers: not practical
  - Arithmetic mean: simple but effective

Training

Environment – Google Cloud Compute Engine

- 6 vCPUs, 30 GB RAM and 1 Nvidia Tesla K80 GPU

Data generation

- Balanced positive and negative training examples
  - the given 3000 binding pairs
  - randomly selected 3000 non-binding pairs with distance <= 10
- For each pair, generate ROI grids for all ligand atoms in parallel
- Split 20% ROIs as validation data

Training: ~ 15 mins, 7 epochs

Testing

- Test the CNN model on
  - the 3000 binding pairs
  - another 3000 random non-binding pairs

- Simulate the final evaluation with randomly selected
  - 400 proteins and 400 ligands
  - 600 proteins and 600 ligands

Results

- Training

Training accuracy	97%
Validation accuracy	94.5%

- Testing

Classification accuracy on 6000 pairs	95%
Accuracy on testing dataset by suggesting 10 candidates	400 proteins and 400 ligands: 100%
	600 proteins and 600 ligands: 98.9%

Takeaways

- CNN is powerful
  - You don’t need to know anything about protein ligand affinity
  - Neither does the neural network
- Better techniques are needed to make it practical and general-purpose
  - How to efficiently do ROI pooling without the strong assumptions?

Thank you