Resolve Hanabi with Neural Network

Chiappini David e Ciruzzi Michele

May 10, 2018

1 Notes

1.1 Input

The necessary input are:

- 1 for the number of lifes [int]
- 1 for the number of hints [int]
- 1 for the number of tiles left in the deck [int]
- 20 for the discarded tiles:
  - 1 for each of the 20 possible couple number-color [int]
- 64 for 3 opponents' hands:
  - 16 for each opponent's hand:
    - 1 for the color of each of the 4 tiles in the hand [int]
    - 1 for the number of each of the 4 tiles in the hand [int]
    - 2 for each of the 4 tiles in the hand to register if owner know color and/or number [bool]
- 8 for the player's hand:
  - 1 for the color of each of the 4 tiles in the hand (0 if unknown) [int]
  - 1 for the number of each of the 4 tiles in the hand (0 if unknown) [int]
- 4 for the tiles on the stacks [int]

For a total of 99 input.
Stacks and hands use progressive integer to input the different numbers and colors of tiles. If the training will show often misunderstandings of number or color I suggest to split the integer input in a greater number of boolean input.

1.2 Output

The necessary output are:

- 4 to discard each of tiles in the hand [probability]
- 4 to play each of tiles in the hand [probability]
- 27 to give a hint to an opponent:
  - 5 numbers for each of the 3 opponents [probability]
  - 4 colors for each of the 3 opponents [probability]

For a total of 35 output.
Eventually each of this output could be a boolean input for the next player to keep memory of recent plays.
E.g. with 116 additional input we give to the network memory of the last play of each player (we manage draw and discard as single action instead of four for this purpose).

1.3 Other Choices for Neural Network

The first choice necessary to characterize neural network is the function that characterizes neurons.
An obvious choice is the sigmoide $\sigma(z) = \frac{1}{1+e^{-z}}$ but for output neurons we should prefer a normalized function like the softmax $\sigma_m(z_j) = \frac{e^{z_j}}{\sum_i e^{z_i}}$.
In both case we have, if i is a neuron of $L-1$ layer and j is our neuron in L layer, $z_j = \sum_i w_{ij} z_i + b_{ij}$, with w the "weight" of the link and b the associates bias.
An interesting choice for the error function for each iteration is $C = \left|p - \frac{s}{20}\right|^{\alpha n}$, where s is the score achieved, n is the distance (in number of moves) from the end of the match, α a parameter and p the probability of the move (i.e. the values of the output neuron chosen).
This choice allows us to not take any assumption about what is a good move and, at the same time, to give a stronger feedback to the last moves (which we can suppose are more linked with the final result) and weaker to the farther ones.
The idea behind the function is that we are more likely to choose a move if it grants us a better result, but (except the case in which it grants us the perfect match) we want to preserve a possibility of deviation to improve.
Our source suggest the use of a mean of cost function on a sample of the training set, but I'd refuse the guess.
Also propose to update weight and bias as follow:

$$w'_j = w_j - \eta \frac{\partial C}{\partial w_j}$$

$$b'_j = b_j - \eta \frac{\partial C}{\partial b_j}$$

1.4 Algebra

Hypothesize two hidden layer.
So we have

- 99 a_μ input
- 110 (arbitrary) sigmoid neurons of first hidden layer

$$h^1_\lambda = \sigma\left(w^{1\mu}{}_\lambda a_\mu + b^1{}_\lambda\right) = \left(1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right)\right)^{-1}$$

- 110 (still arbitrary) sigmoid neurons of second hidden layer

$$\begin{aligned} h^2_\nu &= \sigma\left(w^2{}_\nu{}^\lambda h^1_\lambda + b^2{}_\nu\right) = \left(1 + exp\left(-w^2{}_\nu{}^\lambda h^1_\lambda - b^2{}_\nu\right)\right)^{-1} = \\ &= \left(1 + exp\left(-w^2{}_\nu{}^\lambda \cdot \sigma\left(w^{1\mu}{}_\lambda a_\mu + b^1{}_\lambda\right) - b^2{}_\nu\right)\right)^{-1} = \\ &= \left(1 + exp\left(-w^2{}_\nu{}^\lambda \left(1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right)\right)^{-1} - b^2{}_\nu\right)\right)^{-1} \end{aligned}$$

- 35 output softmax neurons

$$\begin{aligned} o_{\hat\rho} &= \sigma_m\left(w^{o\nu}{}_{\hat\rho} h^2_\nu + b^o{}_{\hat\rho}\right) = \frac{exp(w^{o\nu}{}_{\hat\rho} h^2_\nu + b^o{}_{\hat\rho})}{\sum_\rho exp(w^{o\nu}{}_\rho h^2_\nu + b^o{}_\rho)} = \\ &= \frac{exp(w^{o\nu}{}_{\hat\rho} \cdot \sigma\left(w^2{}_\nu{}^\lambda h^1_\lambda + b^2{}_\nu\right) + b^o{}_{\hat\rho})}{\sum_\rho exp(w^{o\nu}{}_\rho \cdot \sigma\left(w^2{}_\nu{}^\lambda h^1_\lambda + b^2{}_\nu\right) + b^o{}_\rho)} = \\ &= \frac{exp\left(w^{o\nu}{}_{\hat\rho}\left(1 + exp\left(-w^2{}_\nu{}^\lambda \left(1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right)\right)^{-1} - b^{2\lambda}\right)\right)^{-1} + b^o{}_{\hat\rho}\right)}{\sum_\rho exp\left(w^{o\nu}{}_\rho\left(1 + exp\left(-w^2{}_\nu{}^\lambda (1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right))^{-1} - b^{2\lambda}\right)\right)^{-1} + b^o{}_\rho\right)} \end{aligned}$$

Out cost function will become

$$\begin{aligned} C &= \left|o_{\hat\rho} - \frac{s}{20}\right|^{\alpha n} = \left|\sigma_m\left(w^{o\nu}{}_{\hat\rho} \cdot \sigma\left(w^2{}_\nu{}^\lambda \cdot \sigma\left(w^{1\mu}{}_\lambda a_\mu + b^1{}_\lambda\right) + b^2{}_\nu\right) + b^o{}_{\hat\rho}\right) - \frac{s}{20}\right|^{\alpha n} = \\ &= \left|\frac{exp\left(w^{o\nu}{}_{\hat\rho}\left(1 + exp\left(-w^2{}_\nu{}^\lambda \left(1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right)\right)^{-1} - b^2{}_\nu\right)\right)^{-1} + b^o{}_{\hat\rho}\right)}{\sum_\rho exp\left(w^{o\nu}{}_\rho\left(1 + exp\left(-w^2{}_\nu{}^\lambda (1 + exp\left(-w^{1\mu}{}_\lambda a_\mu - b^1{}_\lambda\right))^{-1} - b^2{}_\nu\right)\right)^{-1} + b^o{}_\rho\right)} - \frac{s}{20}\right|^{\alpha n} \end{aligned}$$

So, we need to compute $\frac{\partial C}{\partial w^1}$, $\frac{\partial C}{\partial w^2}$, $\frac{\partial C}{\partial w^o}$, $\frac{\partial C}{\partial b^1}$, $\frac{\partial C}{\partial b^2}$, $\frac{\partial C}{\partial b^o}$ in order to determinate the variation of weights and bias.
Remember that $\sigma'(x) = \sigma(x)(1 - \sigma(x))dx$ and $\sigma'_m(x) = \sigma_m(x)(1 - \sigma_m(x))dx$ and consider that we can extend ∂C in $o = \frac{s}{20}$ assigning the value 0, we can

start computing $dC = \frac{\alpha n}{o - \frac{s}{20}} C(o) do$.

Furthermore we can labeling $z^o = w^{o\nu}_{\hat{\rho}} h^2_\nu + \sum_\nu b^{o\nu}_{\hat{\rho}}$, $z_2 = w^{2\lambda}_{\ \ \nu} h^1_\lambda + \sum_\lambda b^{2\lambda}_{\ \ \nu}$ and $z_1 = w^{1\lambda}_{\ \ \nu} a_\lambda + \sum_\lambda b^{1\lambda}_{\ \ \nu}$, such that $o_{\hat{\rho}} = \sigma_m(z^o)$, $h^2 = \sigma(z^2)$ and $h^1 = \sigma(z^1)$.
So follow that:

$$\frac{\partial C}{\partial b^o} = \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial b^o} = \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o))$$

$$\frac{\partial C}{\partial w^{o\mu}} = \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial w^{o\mu}} = \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o)) h^2{}_\mu = \frac{\partial C}{\partial b^o} h^2{}_\mu$$

$$\begin{aligned}
\frac{\partial C}{\partial b^2} &= \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial b^2} = \\
&= \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o))(w^{o\mu} \sigma(z^2{}_\mu)(1 - \sigma(z^2{}_\mu)) = \\
&= \frac{\partial C}{\partial b^o} (w^{o\mu} \sigma(z^2{}_\mu)(1 - \sigma(z^2{}_\mu))
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C}{\partial w^{2\nu}} &= \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial w^{2\nu}} = \\
&= \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o))(w^{o\mu} \sigma(z^2{}_\mu)(1 - \sigma(z^2{}_\mu)) h^1{}_\nu = \frac{\partial C}{\partial b^2} h^1{}_\nu
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C}{\partial b^1} &= \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial b^1} = \\
&= \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o))(w^{o\mu} \sigma(z^2{}_\mu)(1 - \sigma(z^2{}_\mu))(w^{2\nu} \sigma(z^1{}_\nu)(1 - \sigma(z^1{}_\nu)) = \\
&= \frac{\partial C}{\partial b^2} (w^{2\nu} \sigma(z^1{}_\nu)(1 - \sigma(z^1{}_\nu))
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C}{\partial w^{1\tau}} &= \frac{\alpha n}{o - \frac{s}{20}} C(o) \frac{\partial \sigma_m(z^o)}{\partial w^{1\tau}} = \\
&= \frac{\alpha n}{o - \frac{s}{20}} C(o) \sigma_m(z^o)(1 - \sigma_m(z^o))(w^{o\mu} \sigma(z^2{}_\mu)(1 - \sigma(z^2{}_\mu))(w^{2\nu} \sigma(z^1{}_\nu)(1 - \sigma(z^1{}_\nu)) a_\tau = \\
&= \frac{\partial C}{\partial b^1} a_\tau
\end{aligned}$$