En raison des résultats obtenus sur la base de données de CMU et la nôtre, j’ai effectué de nouvelles recherches sur les réseaux de neurones. Dans un premier temps, les résultats obtenus à chaque nouvel entraînement sont différents principalement à cause des poids attribués sur chaque synapse. L’attribution des ces poids est réalisé grâce à la fonction d’initialisation « GlorotUniform » (Xavier Glorot) laquelle permet une descente de gradient plus rapide et adapté aux bases de données peu référencées. Le sur-apprentissage qui est un problème courant avec l’apprentissage de mouvement grâce aux couches LSTM, peut être évité grâce à « Recurrent\_Dropout » (paramètre de la couche LSTM) ou à la couche Dropout. J’utilise plutôt cette première option car le « l’oubli » des données se fait au cours du temps (horizontalement) contrairement à la couche (verticalement). Enfin, la fonction de perte « sparsecategoricalcrossentropy » est utilisée sur des bases de données importantes, dans notre cas, deux procédures sont plus adaptées : meansquarrederror et categorical\_crossentropy sont alors utilisé et procure un taux de perte faible. La couche BatchNormalisation permet aussi d’améliorer les prédictions en maintenant la sorti proche de 0 et l’écart-type proche de 1.

Avec quelques ajustements des autres paramètres (neurones, epochs, learning rate, batch size), un taux de réussite de 100% est obtenu avec notre base de données et celle de l’université de Carnegie Mellon (deux mouvements pour chaque base sont considérés pour le moment).

D’autres solutions sont à l’étude, telles que recentrer le sujet dans la matrice (prendre la vertèbre principale comme point de repère) et utilisé une couche de convolution (convolution temporel).