

【要点まとめ】 学習率は大きすぎても小さすぎても、最適な解を見つけられない。（ここは復習） NNに最適な学習をさせるために、幾つかの学習率最適化手法がある。 ・ Adam（現時点でかなりホット！） ・ Adagrad ・ モメンタム ・ RMSProp

【実装演習】

```
# モメンタム
self.v[key] = self.momentum * self.v[key] - self.learning_rate * grad[key]
params[key] += self.v[key]

# AdaGrad
self.h[key] = np.zeros_like(val)
self.h[key] += grad[key] * grad[key]
params[key] -= self.learning_rate * grad[key] / (np.sqrt(self.h[key]) + 1e-7)

# RMSProp
self.h[key] *= self.decay_rate
self.h[key] += (1 - self.dacay_rate) * grad[key] * grad[key]
params[key] -= self.learning_rate * grad[key] / (np.sqrt(self.h[key]) + 1e-7)

# Adam
self.m[key] = np.zeros_like(val)
self.v[key] = np.zeros_like(val)
self.m[key] += (1 - self.beta1) * (grad[key] - self.m[key])
self.v[key] += (1 - self.beta2) * (grad[key] ** 2 - self.v[key])
params[key] -= self.learning_rate * self.m[key] / (np.sqrt(self.v[key]) + 1e-7)
```

※ 参考資料：2_4_optimizer.ipynb

【実装演習考察】 AdaGrad → RMSProp → Adam の進化の過程がコードにも現れているように感じました。基本的にはAdamを使うべきと伺いましたが、他の手法も知ることによってその理解も深まると思いますので 実装する際はそれらを意識するようにします。