Capstone Design

2018년 2학기

2018년 11월 8일

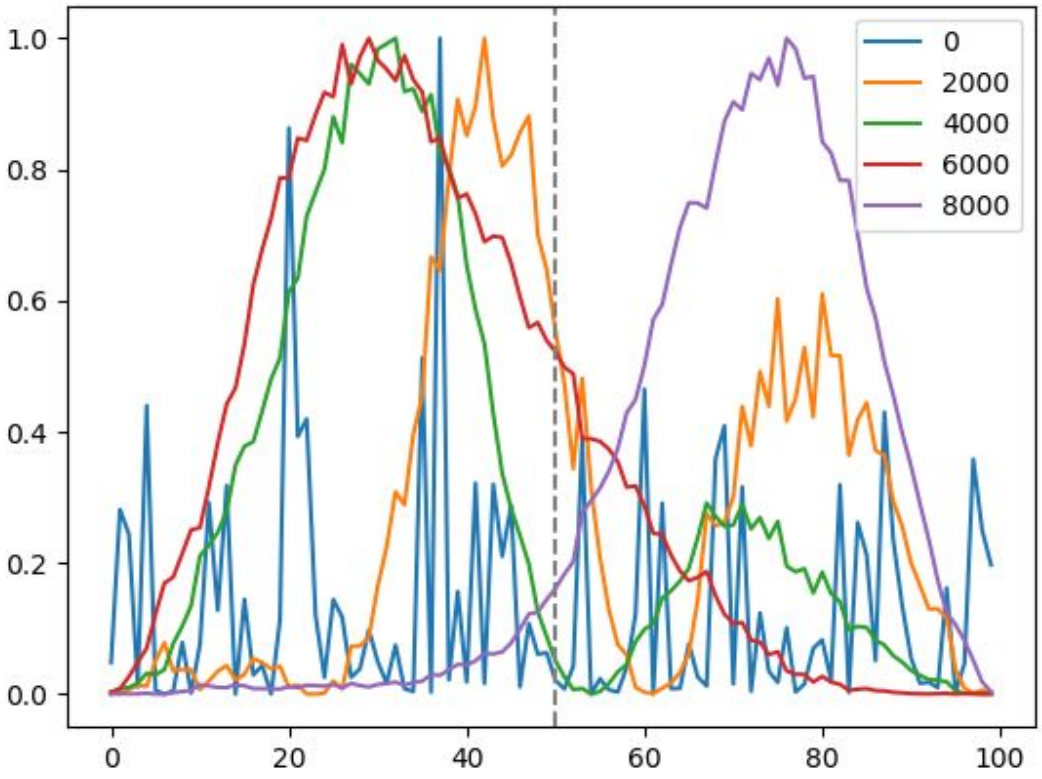
TensorFlow Optimizer의 특징과 현재 Wavefunction Model에 기반한 성능 평가

Gradient Descent Optimizer

- Neural Network의 가장 기본적인 학습 방법
- Gradient Descent: 기울기가 줄어드는 방향으로 일정 크기만큼 이동

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta_t)$$

  - θ parameter
  - $J(\theta)$ cost function
  - η learning rate
- learning rate 값에 따라 local minima에 빠지거나 발산(overshooting)
- 전체 train data set에 대한 계산을 마친 후 parameter 업데이트 ➡ 계산 양 많음



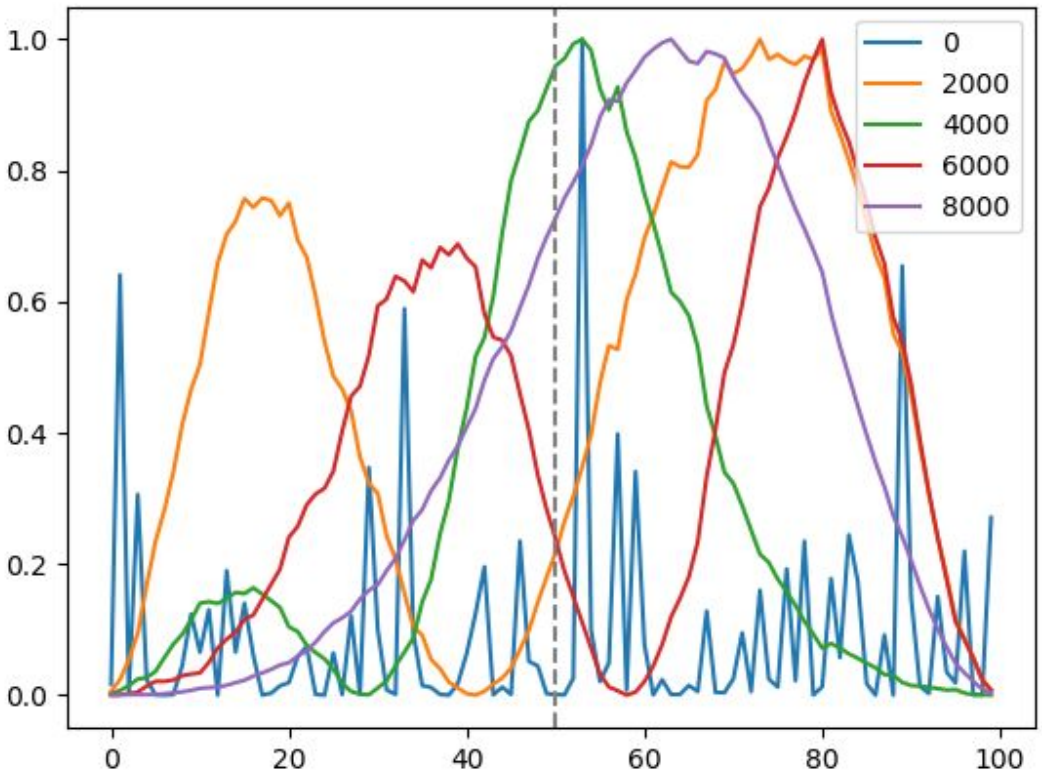
Momentum Optimizer

- Gradient Descent Optimizer를 관성의 원리를 통해 보완

$$\theta_{t+1} = \theta_t - v_{t+1}$$

$$v_{t+1} = \gamma v_t + \eta \nabla_\theta J(\theta_t)$$

  - γ momentum constant (일반적으로 약 0.9)
  - v_t 이동 벡터
- local minima 문제 해결에 도움
- parameter를 갱신할 때마다 이동 벡터도 함께 갱신 ➡ 메모리 사용량 두 배



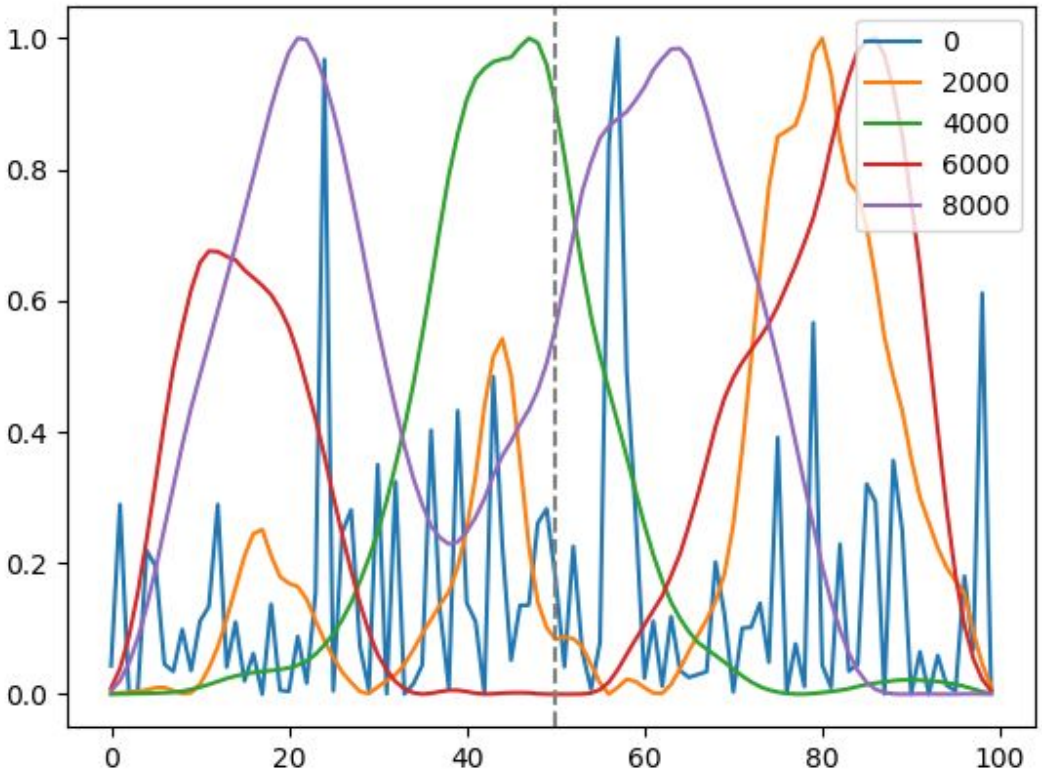
Adagrad Optimizer

- 변화가 없을수록 크게 이동, 변화가 클수록 적게 이동

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_{t+1}+\varepsilon}} \cdot \nabla_\theta J(\theta_t)$$

$$G_{t+1} = G_t + (\nabla_\theta J(\theta_t))^2$$

  - G_t learning rate 조정 계수
  - ε $G_t = 0$ 일 때 분모가 0이 되는 것을 방지 ($10^{-8} \sim 10^{-4}$)
- 빠르게 최적화 위치에 도달
- G_t 가 단조 증가하므로 iteration이 커지면 learning rate가 줄어듦



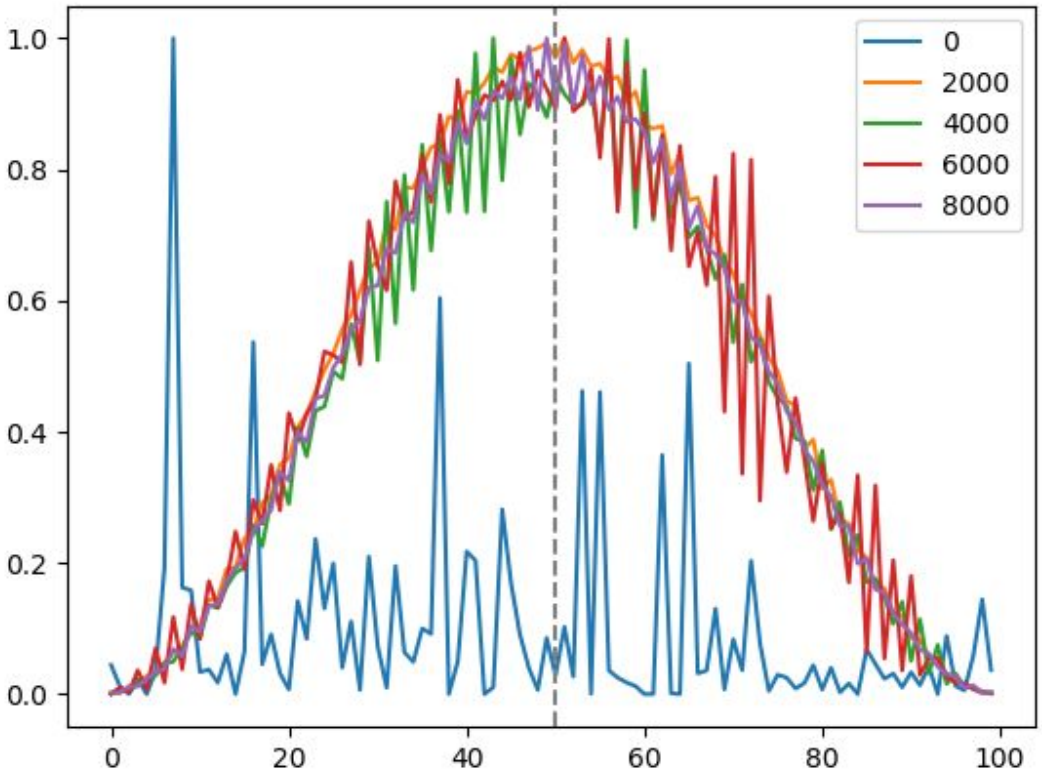
RMSProp Optimizer

- EMA를 도입하여 Adagrad를 보완

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_{t+1}+\varepsilon}} \cdot \nabla_\theta J(\theta_t)$$

$$G_{t+1} = \gamma G_t + (1-\gamma)(\nabla_\theta J(\theta_t))^2$$

- G_t에 exponential moving average를 사용, iteration이 커져도 learning rate가 무조건적으로 줄어드는 것을 방지
- momentum optimization 기법을 함께 적용 가능



Adadelta Optimizer

- 변화하는 learning rate와 EMA를 도입하여 Adagrad를 보완

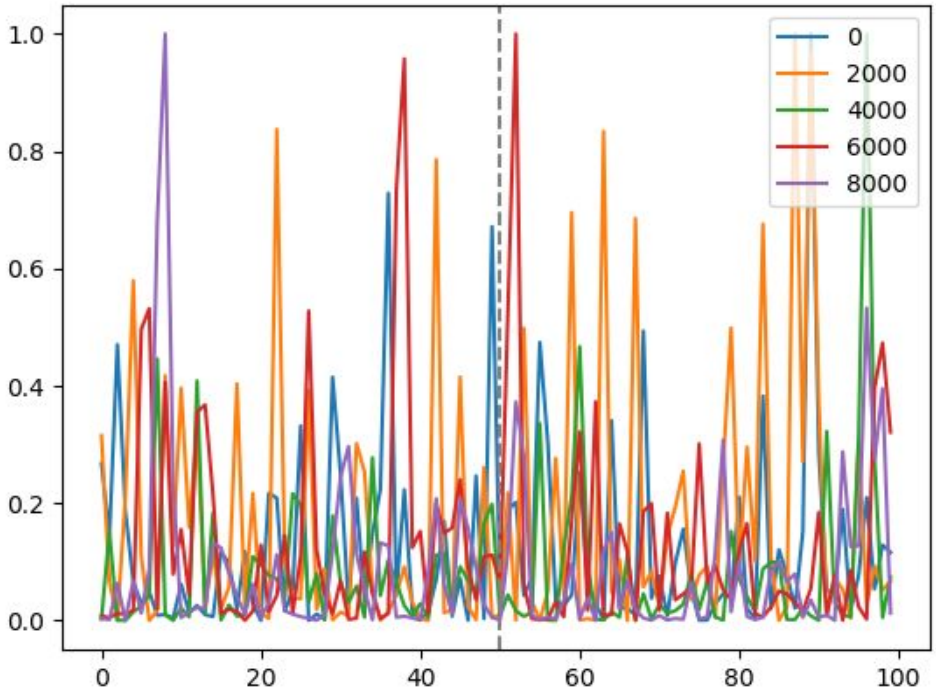
$$\theta_{t+1} = \theta_t - \Delta_\theta$$

$$\Delta_\theta = \frac{\sqrt{s+\varepsilon}}{\sqrt{G+\varepsilon}} \cdot \nabla_\theta J(\theta_t)$$

$$s_{t+1} = \gamma s_t + (1-\gamma)\Delta_\theta^2$$

$$G_{t+1} = \gamma G_t + (1-\gamma)(\nabla_\theta J(\theta_t))^2$$

- 고정된 learning rate를 쓰는 것이 아니므로 Adagrad와 견주어 느림
- error 값의 변화가 천천히 반영되므로 최적화 성능이 낮음



Adam Optimizer

- RMSProp과 Momentum의 결합으로 현재 NN에서 가장 많이 사용됨

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \hat{m}_t$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$m_{t+1} = \beta_1 m_t + (1-\beta_1)\nabla_\theta J(\theta_t)$$

$$v_{t+1} = \beta_2 v_t + (1-\beta_2)(\nabla_\theta J(\theta_t))^2$$